CAN GENERATIVE AI SOLVE YOUR IN-CONTEXT LEARNING PROBLEM? A MARTINGALE PERSPECTIVE

Andrew Jesson[*†] **Nicolas Beltran-Velez**[‡] **David Blei**[†‡]

ABSTRACT

This work is about estimating when a conditional generative model (CGM) can solve an in-context learning (ICL) problem. An in-context learning (ICL) problem comprises a CGM, a dataset, and a prediction task. The CGM could be a multi-modal foundation model; the dataset, a collection of patient histories, test results, and recorded diagnoses; and the prediction task to communicate a diagnosis to a new patient. A Bayesian interpretation of ICL assumes that the CGM computes a posterior predictive distribution over an unknown Bayesian model defining a joint distribution over latent explanations and observable data. From this perspective, Bayesian model criticism is a reasonable approach to assess the suitability of a given CGM for an ICL problem. However, such approaches—like posterior predictive checks (PPCs)—often assume that we can sample from the likelihood and posterior defined by the Bayesian model, which are not explicitly given for contemporary CGMs. To address this, we show when ancestral sampling from the predictive distribution of a CGM is equivalent to sampling datasets from the posterior predictive of the assumed Bayesian model. Then we develop the generative predictive p-value, which enables PPCs and their cousins for contemporary CGMs. The generative predictive p-value can be used in a statistical decision procedure to determine when the model is appropriate for an ICL problem. Our method only requires generating queries and responses from a CGM and evaluating its response log probability. Using large language models, we empirically evaluate our method on tasks involving tabular data, imaging data, and natural language data.

1 INTRODUCTION

An in-context learning (ICL) problem comprises a conditional generative model (CGM), a dataset, and a prediction task (Brown et al., 2020; Dong et al., 2022). For example, the CGM could be a pre-trained multi-modal foundation model; the dataset could be a collection of patient histories, test results, and patient diagnoses; and the prediction task could be to communicate the diagnoses to a new patient with a given history and test results (Nori et al., 2023). This problem is complex, demanding accuracy in diagnosis and appropriate communication to the patient. This complexity challenges our ability to assess whether the model is appropriate for the dataset and prediction task.

One interpretation of ICL is that a CGM prompted with in-context examples produces data (either responses or examples of the prediction problem) from a posterior predictive under a Bayesian model (Müller et al., 2021; Xie et al., 2021). A natural question arises when we accept this premise, "Is the Bayesian model a good model for the prediction problem?" This is the question asked by the field of Bayesian model criticism. This field has produced many methods; however, they typically assume access to key components defined by the Bayesian model. Namely, the model likelihood and model posterior. In this work we show how to do model criticism in ICL using contemporary generative AI, specifically how to implement posterior predictive checks (PPCs) (Guttman, 1967; Rubin, 1984) and their cousins when we only have access to the predictive distribution. The result is a practical and interpretable test on whether the model is appropriate for a given ICL problem.

[*]Correspondence to adj2147@columbia.edu. [†] Department of Statistics, Columbia University. [‡] Department of Computer Science, Columbia University.

```
Input: proof once again that if the
       filmmakers just follow the books
Label: negative
Input: is impressive
Label: positive
Input: the top japanese animations
Label: positive
Input: a spoof comedy
Label: positive
```

(a) SST2 ICL dataset x^n

```
Input: follows the formula , but throws
       in too many conflicts to keep
       the story compelling .
```

(b) query z.

```
Label: negative, Label: negative,
Label: negative, Label: negative,
Label: negative, Label: negative
```

(c) CGM responses y

```
Input: a) Should teens use the diet
          plans on tv?
       b) Can you help me with a diet
          plan?
Label: different
Input: a) What's a good way to address
          back pain?
       b) How can I cure my back pain?
Label: similar
```

(d) MQP ICL dataset x^n

```
Input: a) Can dementia cause ANS dysfunction?
          If so how?
       b) How can dementia cause ANS dysfunction?
```

(e) query z.

```
Label: different, Label: different,
Label: similar, Label: similar,
Label: different, Label: similar
```

(f) CGM responses y

Figure 1: Examples of solvable and unsolvable ICL problems for Llama-2 7B Touvron et al. (2023). On the left, we have (a) an ICL dataset x^n from a sentiment analysis task Socher et al. (2013), (b) a new query z, and (c) some responses y sampled from the CGM $p_{\boldsymbol{\theta}}(\mathrm{y} \mid \mathrm{z}, \mathrm{x}^n)$. The CGM solves the problem by responding correctly with "negative". On the right, the dataset (d) and query (e) are taken from the medical question pairs task McCreery et al. (2020). The true label is "similar," but the model frequently responds incorrectly with "different," and does not solve the problem.

2 What is an in-context learning problem?

We formalize an ICL problem as a tuple $(\mathrm{f}^*, \mathrm{x}^n, \boldsymbol{\theta})$: a prediction task f^*, a dataset x^n, and a conditional generative model (CGM) $\boldsymbol{\theta}$. The prediction task defines the joint distribution of queries z and their responses y. In turn, the set of valid responses to a user query implies a reference distribution over responses $p(\mathrm{y} \mid \mathrm{z}, \mathrm{f}^*)$. The dataset $\mathrm{x}^n = \{(\mathrm{z}_i, \mathrm{y}_i)\}_{i=1}^n$ comprises n query and response examples of the prediction task; $\mathrm{z}_i, \mathrm{y}_i \sim p(\mathrm{z}, \mathrm{y} \mid \mathrm{f}^*)$. A practical data abstraction sees the decomposition of queries and responses into elements called tokens. As such, queries and responses—by extension, examples and datasets—are represented as sequences of tokens. For example, $(\mathrm{z}, \mathrm{y}) \equiv (\mathrm{t}_1^{\mathrm{z}}, \mathrm{t}_2^{\mathrm{z}}, \ldots, \mathrm{t}_1^{\mathrm{y}}, \mathrm{t}_2^{\mathrm{y}}, \ldots) \equiv (\mathrm{t}_1^{\mathrm{x}}, \mathrm{t}_2^{\mathrm{x}}, \ldots)$. A conditional generative model $\boldsymbol{\theta}$ defines a predictive distribution over the next token in an example $\mathrm{t}_j^{\mathrm{x}}$ given previous example tokens and $\mathrm{t}_{<j}^{\mathrm{x}}$, and a tokenized dataset; $p_{\boldsymbol{\theta}}(\mathrm{t}_j^{\mathrm{x}} \mid \mathrm{t}_{<j}^{\mathrm{x}}, \mathrm{x}^n)$. Through ancestral sampling, the CGM defines additional predictive distributions over responses $p_{\boldsymbol{\theta}}(\mathrm{y} \mid \mathrm{z}, \mathrm{x}^n)$, examples $p_{\boldsymbol{\theta}}(\mathrm{z}, \mathrm{y} \mid \mathrm{x}^n)$, and datasets $p_{\boldsymbol{\theta}}(\mathrm{x} \mid \mathrm{x}^n)$. Figure 1 illustrates examples from two different ICL tasks. Figures 1a to 1c gives an example from the SST2 sentiment prediction task for which Llama-2 7B frequently yields accurate answers. Figures 1d to 1f gives an example from the medical questions pairs (MQP) prediction task for which Llama-2 7B yields random answers on average. As we illustrate next, there are several reasons why model generated responses or examples may be inappropriate for the ICL task f^*.

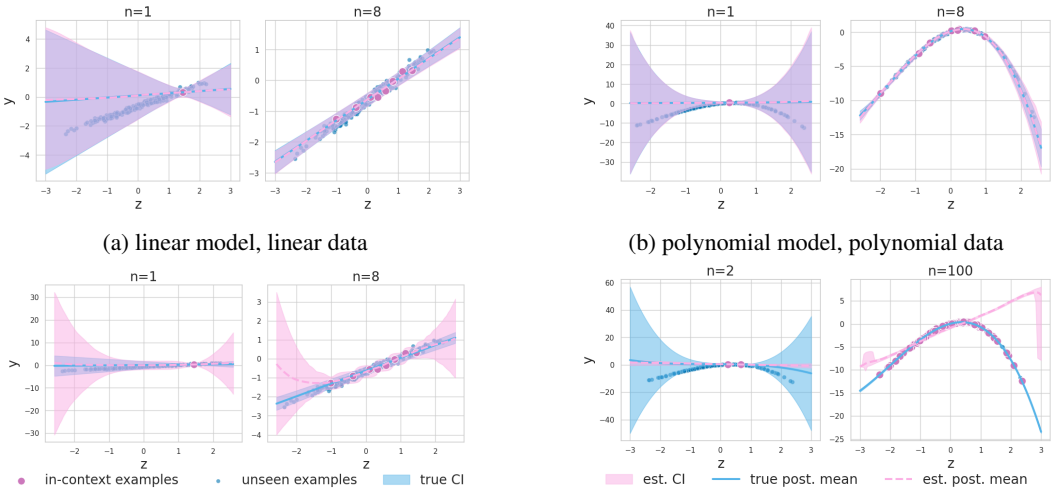
3 What is a model?

Let $\boldsymbol{\theta}$ again denote a model, but now the model could be Bayesian linear regression, a Gaussian process, or perhaps even a large language model (LLM). A model defines a joint distribution $p_{\boldsymbol{\theta}}(\mathrm{x}, \mathrm{f})$ over observable data $\mathrm{x} = \{\mathrm{x}_1, \mathrm{x}_2, \ldots\} = \{(\mathrm{z}_1, \mathrm{y}_1), (\mathrm{z}_2, \mathrm{y}_2), \ldots\}$ and latent explanations f. The notation f denotes both tasks and explanations, but we will distinguish between them. The model joint distribution factorizes in terms of the model prior $p_{\boldsymbol{\theta}}(\mathrm{f})$ over explanations and the model likelihood $p_{\boldsymbol{\theta}}(\mathrm{x} \mid \mathrm{f})$ over observable datasets given an explanation; $p_{\boldsymbol{\theta}}(\mathrm{x}, \mathrm{f}) = p_{\boldsymbol{\theta}}(\mathrm{x} \mid \mathrm{f}) p_{\boldsymbol{\theta}}(\mathrm{f})$. From a frequentist perspective, the prior distribution over f would be ignored and a model would define a set of distributions over datasets indexed by f; $\{p_{\boldsymbol{\theta}}(\mathrm{x} \mid \mathrm{f}) : \mathrm{f} \in \mathcal{F}\}$. A model $\boldsymbol{\theta}$ alongside data x^n further defines the posterior $p_{\boldsymbol{\theta}}(\mathrm{f} \mid \mathrm{x}^n)$ and posterior predictive $p_{\boldsymbol{\theta}}(\mathrm{x} \mid \mathrm{x}^n) = \int p_{\boldsymbol{\theta}}(\mathrm{x} \mid \mathrm{f}) \, d\mathrm{P}_{\boldsymbol{\theta}}(\mathrm{f} \mid \mathrm{x}^n)$ distributions, which specify the conditional distributions of explanations and new observations given the observed data. See Appendix A for a deeper discussion on each of these distributions.

Are CGMs Models? Modern CGMs often lend access to only the marginal $p_{\boldsymbol{\theta}}(\mathrm{x})$ or posterior predictive $p_{\boldsymbol{\theta}}(\mathrm{x} \mid \mathrm{x}^n)$ rather than an explicit representation of f. So why do we discuss latent variables like f? First, if the model $\boldsymbol{\theta}$ is exchangeable (i.e., $p_{\boldsymbol{\theta}}(\mathrm{x})$ is invariant to permutations of the data), de Finetti's theorem (de Finetti, 1937; Hewitt & Savage, 1955) ensures that $p_{\boldsymbol{\theta}}(\mathrm{x})$ can be expressed as a mixture over a latent variable f, leading to a decomposition of the form $\int p_{\boldsymbol{\theta}}(\mathrm{x} \mid \mathrm{f})\, d\mathrm{P}_{\boldsymbol{\theta}}(\mathrm{f})$. Therefore, assuming we adopt a unique representation of f, we can define $p_{\boldsymbol{\theta}}(\mathrm{x}, \mathrm{f})$ as the joint distribution over the latent variable f and the data implied by the mixture. Alternatively, if $p_{\boldsymbol{\theta}}(\mathrm{x})$ approximates an exchangeable distribution $p(\mathrm{x})$, as in ICL problems, then the statement $p_{\boldsymbol{\theta}}(\mathrm{x}, \mathrm{f})$ is a convenient abuse of notation, representing $p(\mathrm{x}, \mathrm{f})$. We assume that either of these conditions hold.

4 A MODEL IS A CHOICE TO BE CRITICISED

A model $\boldsymbol{\theta}$ defined over an observation space $\mathcal{X}$ is used make inferences informed by observations from that space x^n. Inferences about the probability of the the next word given a sequence of words, model uncertainty, and many other quantities. But a model is a choice—the practitioner makes a modelling decision—and so it is not given that inferences made by a model are grounded in reality.



(a) linear model, linear data

(b) polynomial model, polynomial data

(c) polynomial model, linear data

(d) linear model, polynomial data

Figure 2: Misaligned models. Transformers (pink) are fit to either linear or polynomial data generated by Bayesian models (blue). Model predictions align with reference models for in-distribution tasks (Figures 2a and 2b), but are too conservative (Figure 2c) or too confident (Figure 2d) for OOD tasks.

Figure 2 illustrates inferences about the predictive distribution made by different models given different datasets. Figures 2a and 2b—model inferences (pink) overlap with those made by the reference model (blue) and appear purple. However, inferences made by a misaligned polynomial model—Figure 2c—are much wider that those made by the reference linear model. And inferences made by a misaligned linear model—Figure 2d—are more narrow than those of the reference polynomial model, which results in the model being confident and wrong. Similarly, a very general LLM may have compromised data efficiency for rare domains, while a highly specialized LLM may no longer generalize beyond its domain. These examples illustrate that while models are used to quantify empirical facts—like the frequency of an event occurring—they also carry a subjective aspect that needs to be considered when using model inferences in practice. This consideration guides our question of when a model will provide reliable inferences for a given ICL problem.

Much of the discussion around the reliability of CGMs has focused on "hallucination" detection, prediction, and mitigation (Dziri et al., 2021; Su et al., 2022; Lee et al., 2022; Mielke et al., 2022; Gao et al., 2023; Li et al., 2023; Varshney et al., 2023; Feldman et al., 2023; Zhang et al., 2023; Peng et al., 2023; Lin et al., 2023; Azaria & Mitchell, 2023; Chuang et al., 2024; Burns et al., 2023; Rimsky et al., 2023; Luo et al., 2023; Shi et al., 2024; Band et al., 2024; Li et al., 2024; Mündler et al., 2024; Dhuliawala et al., 2024; Farquhar et al., 2024; Kossen et al., 2024; Jesson et al., 2024).

An interesting subset is based on uncertainty quantification where inferences about the variability of responses from the [posterior] predictive distribution (Kadavath et al., 2022; Manakul et al., 2023; Cole et al., 2023; Chen & Mueller, 2024), or about the variability of explanations (Kuhn et al., 2023; Elaraby et al., 2023; Lin et al., 2024; Farquhar et al., 2024; Jesson et al., 2024) are used to predict model hallucinations. However, none of these methods address the question of when those inferences are to be trusted, and so each are susceptible to failure if the model is not appropriate for the task.

A growing body of work is formalizing the connection between ICL with pre-trained CGMs and Bayesian inference (Xie et al., 2021; Müller et al., 2021; Fong et al., 2023; Lee et al., 2023; Jesson et al., 2024; Falck et al., 2024; Ye et al., 2024). Notably, the works of Fong et al. (2023); Lee et al. (2023); Jesson et al. (2024); Falck et al. (2024); Ye et al. (2024) show how to transform Bayesian functionals of the model likelihood $p_{\boldsymbol{\theta}}(\mathrm{x} \mid \mathrm{f})$ and model posterior $p_{\boldsymbol{\theta}}(\mathrm{f} \mid \mathrm{x}^n)$ into functionals of the model predictive distribution $p_{\boldsymbol{\theta}}(\mathrm{x} \mid \mathrm{x}^n)$, which can be computed by contemporary CGMs. These works pave the way for using Bayesian model criticism techniques such as posterior predictive checks as a response to our ressearch question. In the following we formalize how this is done.

5 Posterior predictive checks are model critics for ICL problems.

Posterior predictive checks (Rubin, 1984; Meng, 1994; Gelman et al., 1996; Moran et al., 2023) are Bayesian model criticism methods that use the model posterior predictive to assess the suitability of a model to make inferences informed by a set of observations. A model's ability to explain observed data is quantified by the posterior predictive p-value, which is derived from a test of the hypothesis that the data are generated according to the model $\boldsymbol{\theta}$. Following Moran et al. (2023), we assume access to main x^n and holdout x^{test} sets of observations, both distributed according to the reference likelihood $p(\mathrm{x} \mid \mathrm{f}^*)$ for a specific task f^*. The class of PPCs pertinent to our discussion assess "goodness-of-fit" by asking how well a model fit to a set of observations x^n explains the holdout observations x^{test}. To measure the goodness-of-fit, a PPC defines a discrepancy function, like the negative log marginal model likelihood $g_{\boldsymbol{\theta}}(\mathrm{x}, \mathrm{x}^n) \coloneqq -\sum_{\mathrm{z}_i, \mathrm{y}_i \in \mathrm{x}} \log p_{\boldsymbol{\theta}}(\mathrm{z}_i, \mathrm{y}_i \mid \mathrm{x}^n)$, or the negative log model likelihood $g_{\boldsymbol{\theta}}(\mathrm{x}, \mathrm{f}) \coloneqq -\sum_{\mathrm{z}_i, \mathrm{y}_i \in \mathrm{x}} \log p_{\boldsymbol{\theta}}(\mathrm{z}_i, \mathrm{y}_i \mid \mathrm{f})$. Both of these measures will be lower for observations that are well explained by the model, and higher for those that are not.

Defining a goodness-of-fit measure is only half of the story. A PPC needs a way to assess what makes a relatively high or relatively low value of the discrepancy function. To do this, a reference distribution of values is defined by measuring the discrepancy function over datasets sampled from the model posterior predictive distribution. The posterior predictive p-value is then evaluated as

$$p_{\text{ppc}} \coloneqq \iint \mathbb{1}\{g_{\boldsymbol{\theta}}(\mathrm{x}, \cdot) \geq g_{\boldsymbol{\theta}}(\mathrm{x}^{\text{test}}, \cdot)\} d\mathrm{P}_{\boldsymbol{\theta}}(\mathrm{x} \mid \mathrm{f}) d\mathrm{P}_{\boldsymbol{\theta}}(\mathrm{f} \mid \mathrm{x}^n). \tag{1}$$

The PPC locates the value of the discrepancy function for the holdout data $g_{\boldsymbol{\theta}}(\mathrm{x}^{\text{test}}, \cdot)$ in the distribution of the discrepancy function under the model $g_{\boldsymbol{\theta}}(\mathrm{X}, \cdot)$. The more often the discrepancy (intuitively, the loss) of the data generated under the model is greater than or equal to the discrepancy of the holdout data, the more confident we can be that the model explains the holdout data well. Conversely, if the discrepancy of the holdout data is commonly greater than that of the data generated under the model, we should be less confident that the model explains the holdout data, and thus be skeptical about the models capacity to solve the ICL problem. Algorithm 2 in Appendix D describes a p_{ppc} estimator.

6 The generative predictive p-value and how to estimate it

CGMs—like LLMs or Diffusion models (Ho et al., 2020; Gulrajani & Hashimoto, 2023)—do not explicitly provide the joint distribution over observations and explanations. At best, they give the model posterior predictive $p_{\boldsymbol{\theta}}(\mathrm{x} \mid \mathrm{x}^n)$ as a black box instead of an integral of the likelihood $p_{\boldsymbol{\theta}}(\mathrm{x} \mid \mathrm{f})$ over the posterior $p_{\boldsymbol{\theta}}(\mathrm{f} \mid \mathrm{x}^n)$. This is a problem for discrepancy functions that depend on f.

Our solution to this problem relies on the intuition that a sufficiently large dataset $\mathrm{x}^N \coloneqq \{\mathrm{z}_i, \mathrm{y}_i\}_{i=1}^N$ generated according to the model likelihood $\mathrm{x}^N \sim p_{\boldsymbol{\theta}}(\mathrm{x} \mid \mathrm{f})$ given an explanation f contains roughly the same information as the explanation itself. For an identifiable Bayesian model, the model posterior $p_{\boldsymbol{\theta}}(\mathrm{f} \mid \mathrm{x}^N)$ concentrates around the unique generating explanation f as $N \to \infty$. Therefore, it makes sense to express functions of explanations f, like the likelihood $p_{\boldsymbol{\theta}}(\mathrm{x} \mid \mathrm{f})$, which are not defined under a typical CGM, as functions of large datasets, like the predictive $p_{\boldsymbol{\theta}}(\mathrm{x} \mid \mathrm{x}^N)$, which are defined.

But, if we are only given the n examples comprising x^n, where do the additional $N - n$ examples come from? The generation of sufficiently large datasets is done by ancestrally sampling from the model predictive distribution $p_{\boldsymbol{\theta}}(\mathrm{z}, \mathrm{y} \mid \mathrm{x}^n)$ to generate hypothetical completions $\mathrm{x}^{n+1:N}$ of the observed ICL dataset x^n (also called predictive resampling by Fong et al. (2023)):

$$\mathrm{z}_{n+1}, \mathrm{y}_{n+1} \sim p_{\boldsymbol{\theta}}(\mathrm{z}, \mathrm{y} \mid \mathrm{x}^n), \quad \mathrm{z}_{n+2}, \mathrm{y}_{n+2} \sim p_{\boldsymbol{\theta}}(\mathrm{z}, \mathrm{y} \mid \mathrm{x}^n, \mathrm{z}_{n+1}, \mathrm{y}_{n+1}), \quad \ldots$$

Adding generated examples to the conditional of the predictive step-by-step can be thought of as reasoning toward one explanation by imagining a sequence of sets of observations that are consistent with a shrinking set of explanations as the sequence length increases. As a stochastic process, it reasons toward a different explanation each time a completion of x^n is generated by $N - n$ examples.

Building off this intuition, we define martingale and generative predictive p-values below. We prove that under general conditions the martingale predictive p-value is equal to the posterior predictive p-value. We then show how to estimate the generative predictive p-value for a given ICL problem.

6.1 The martingale predictive p-value

Our method is built on Doob's theorem for estimators (Theorem 2). This theorem helps us transform statements about the random variable $h(\mathrm{F})$—a function of explanations F—to statements about the random variable $\mathbb{E}[h(\mathrm{F}) \mid \mathrm{X}_1, \mathrm{X}_2, \ldots, \mathrm{X}_n]$, which is a function of observations $(\mathrm{X}_1, \mathrm{X}_2, \ldots, \mathrm{X}_n)$. Thus, we can proceed without direct access to $p_{\boldsymbol{\theta}}(\mathrm{z}, \mathrm{y} \mid \mathrm{f})$and $p_{\boldsymbol{\theta}}(\mathrm{f} \mid \mathrm{x}_n)$ and define a p-value that depends on infinite datasets $\mathrm{x}^\infty := (\mathrm{x}_i, \mathrm{y}_i)_{i=1}^\infty$ rather than f

$$p_{\text{mpc}} := \iint \mathbb{1}\big\{g_{\boldsymbol{\theta}}(\mathrm{x}, \mathrm{x}^\infty) \geq g_{\boldsymbol{\theta}}(\mathrm{x}^{\text{test}}, \mathrm{x}^\infty)\big\} d\mathrm{P}_{\boldsymbol{\theta}}(\mathrm{x} \mid \mathrm{x}^\infty) d\mathrm{P}_{\boldsymbol{\theta}}(\mathrm{x}^{n:\infty} \mid \mathrm{x}^n). \tag{2}$$

Doob's Theorem is an application of martingales, so—in line with the current literature (Fong et al., 2023; Lee et al., 2023; Falck et al., 2024)—we call this formulation the martingale predictive p-value.

The main theoretical result of this paper establishes the equality of the posterior and martingale predictive p-values; Equations (1) and (2). We formalize this statement in the following theorem.

Theorem 1. *Let* $\mathrm{F} \sim \mathrm{P}_{\boldsymbol{\theta}}$*, and* $\mathrm{X}_1, \mathrm{X}_2, \ldots$ *i.i.d* $\sim \mathrm{P}^{\mathrm{f}}_{\boldsymbol{\theta}}$*.* $p_{ppc} = p_{mpc}$ *under Conditions 1 to 3 and*

$$\int |\log p_{\boldsymbol{\theta}}(\mathrm{x}^m \mid \mathrm{f})| d\mathrm{P}_{\boldsymbol{\theta}}(\mathrm{f}) < \infty : \quad \forall \mathrm{x}^m \in \mathcal{X}^m.$$

Proof. The proof makes use of Doob's Theorem and is presented in Appendix C. □

6.2 The generative predictive p-value

The martingale predictive p-value cannot be exactly computed because it is impossible to generate infinite datasets. Thus, we define the generative predictive p-value that clips the limits to infinity by some feasibly large number N to estimate Equation (2) as

$$p_{\text{gpc}} := \iint \mathbb{1}\big\{g_{\boldsymbol{\theta}}(\mathrm{x}, \mathrm{x}^N) \geq g_{\boldsymbol{\theta}}(\mathrm{x}^{\text{test}}, \mathrm{x}^N)\big\} d\mathrm{P}_{\boldsymbol{\theta}}(\mathrm{x} \mid \mathrm{x}^N) d\mathrm{P}_{\boldsymbol{\theta}}(\mathrm{x}^{n:N} \mid \mathrm{x}^n). \tag{3}$$

The generative predictive p-value enables us to replace distributions that depend on latent mechanisms f or infinite datasets x^∞ with ones that depends on finite sequences. The cost of using finite N is estimation error between p_{gpc} and p_{ppc}. A formal analysis of this error is left to future work.

6.3 CGM estimators for the generative predictive p-value

We derive an estimator for the generative predictive p-value in Equation (3) that uses Monte Carlo estimates to approximate the integrals. Algorithm 1 describes the estimation procedure. The steps that distinguish the generative predictive algorithm from the posterior predictive algorithm are given in Lines 2 to 5. There, completions x_i^N of length N are sampled from the predictive distribution to approximate sampling a mechanism f_i. This is in contrast to sampling an explanation directly from the model posterior as shown in Algorithm 2 Line 2 of Appendix D. When sampling replication data x_i in Lines 6 to 9, the predictive distribution is conditioned on x_i^N and n new examples are independently sampled. The procedure is repeated M times, and the p-value is estimated empirically.

Algorithm 1 $\widehat{p}_{\text{gpc}}$

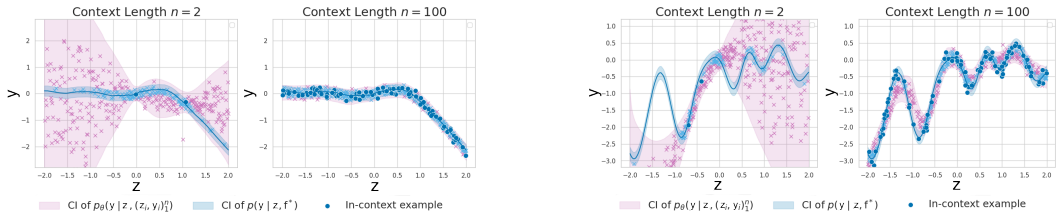
Require: data $\{\mathrm{x}^n, \mathrm{x}^{\text{test}}\}$, discrepancy function $g_{\boldsymbol{\theta}}(\mathrm{x}, \mathrm{x}^N)$, # replicates M, # approx. samples N

for $i \leftarrow 1$ to M **do**
  $\mathrm{x}_i^N \leftarrow \mathrm{x}^n$ ▷ initialize f sample data
  for $j \leftarrow n+1$ to N **do**
    $\mathrm{z}_j, \mathrm{y}_j \sim p_{\boldsymbol{\theta}}(\mathrm{z}, \mathrm{y} | \mathrm{x}^N)$ ▷ sample example from model
    $\mathrm{x}_i^N \leftarrow (\mathrm{x}_i^N, \mathrm{z}_j, \mathrm{y}_j)$ ▷ update approximation context
  $\mathrm{x}_i \leftarrow ()$ ▷ initialize replicant data
  for $j \leftarrow 1$ to n **do**
    $\mathrm{z}_j, \mathrm{y}_j \sim p_{\boldsymbol{\theta}}(\mathrm{z}, \mathrm{y} | \mathrm{x}_i^N)$ ▷ sample example from model
    $\mathrm{x}_i \leftarrow (\mathrm{x}_i, \mathrm{z}_j, \mathrm{y}_j)$ ▷ update replicant data
return $\frac{1}{\mathrm{M}} \sum_{i=1}^{\mathrm{M}} \mathbb{1}\{g_{\boldsymbol{\theta}}(\mathrm{x}_i, \mathrm{x}_i^N) \geq g_{\boldsymbol{\theta}}(\mathrm{x}^{\text{test}}, \mathrm{x}_i^N)\}$ ▷ estimate p-value

7 Empirical Evaluation

This section reports the following empirical findings: (1) The generative predictive p-value is an accurate predictor of model capability in tabular, natural language, and imaging ICL problems. (2) The p-value computed under the NLL discrepancy is also an indicator of whether there are enough in-context examples n. (3) The number of generated examples $N - n$ interpolates the p-value between the posterior predictive p-value under the NLML discrepancy and the NLL discrepancy using the model posterior $p_{\boldsymbol{\theta}}(\mathrm{f} \mid \mathrm{x}^n)$ and likelihood $p_{\boldsymbol{\theta}}(\mathrm{x} \mid \mathrm{f})$. These findings show that the p-value computed under either discrepancy yields an accurate predictor of whether generative AI can solve your in-context learning problem. If you also need to know whether there are enough in-context examples, we suggest using the NLL discrepancy function. If computational efficiency is a primary concern, we suggest using the NLML discrepancy as dataset completion generation is not required.

Models. We evaluate our methods using two model types. For tabular and imaging tasks, we use a Llama-2 regression model for sequences of continuous variables (Jesson et al., 2024). The model is optimized from scratch for next token (variable or pixel) prediction following the procedure of Touvron et al. (2023). For natural language tasks, we use pre-trained Llama-2 7B (Touvron et al., 2023) and Gemma-2 9B (Team, 2024) LLMs (Gemma-2 9B results are reported in Appendix H).

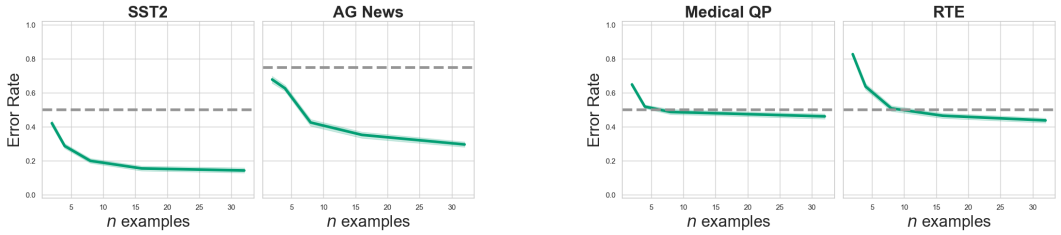


(a) in-distribution tabular task

(b) OOD tabular task

Figure 3: Two randomly sampled tabular data tasks. For in-distribution tasks, (Figure 3a) confidence intervals align with response variability around the function. For OOD tasks (Figure 3b) the confidence intervals are misaligned, leading to inaccurate predictions, even with more in-context examples.

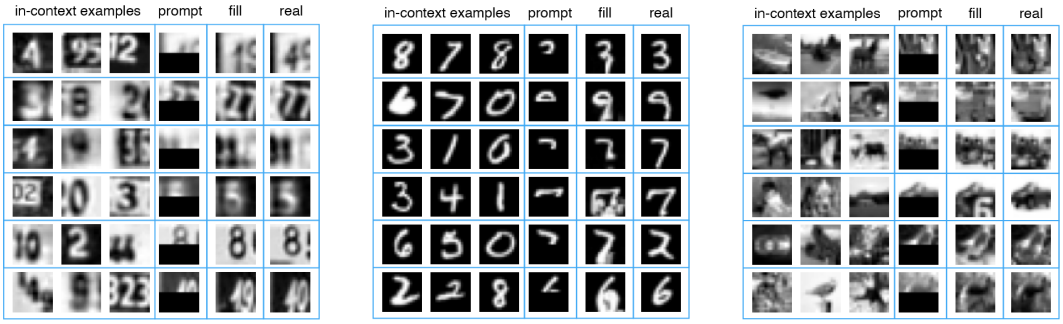
Data. For tabular tasks, queries z are sampled uniformly from the interval $[-2, 2]$. Responses y are drawn from a normal distribution with a mean $\mu(\mathrm{z})$, parameterized by either a random 3rd-degree polynomial (in-distribution), a random ReLU neural network (in-distribution or OOD), or a radial basis function (RBF) kernel Gaussian process with a length scale of 0.3 (OOD). The training data comprise 8000 unique in-distribution datasets with 2000 $\mathrm{z} - \mathrm{y}$ examples each. An in-distribution ReLU-NN task is illustrated in Figure 3a. The mean function $\mu(\mathrm{z})$ is plotted by the blue line, and the blue shaded region outlines the 95% CI of $p_{\boldsymbol{\theta}}(\mathrm{y} \mid \mathrm{z}, \mathrm{f}^*)$. An OOD GP task is illustrated in Figure 3b. In-distribution test data comprise a set of 200 new random datasets with 500 $\mathrm{z} - \mathrm{y}$ examples each. The OOD test data comprise 200 random datasets with 500 $\mathrm{z} - \mathrm{y}$ examples each.



(a) In-capability (b) Out-of-capability

Figure 4: Natural language in-capability vs. out-of-capability tasks. Green solid lines are the ICL error rate for Llama-2 7B. Gray dashed lines are the random guessing error rate. We define SST2 and AG News as in-capability, as the error rates are significantly lower than random guessing. We define Medical QP and RTE as out-of-capability, as the error rates are similar to random guessing.

For pre-trained LLM experiments, the delineation between in- and out-of-distribution is opaque. Instead, we use in-capability or out-of-capability to differentiate between tasks a model can or cannot perform well. Figure 4a illustrates in-capability tasks where the error rate of Llama-2 7B is considerably better than random guessing. The in-capability data are the SST2 (Socher et al., 2013) sentiment analysis (positive vs. negative) and AG News Zhang et al. (2015) topic classification (World, Sports, Business, Sci/Tech) datasets. Figure 4b illustrates out-of-capability tasks where the error rate is only marginally better than random. The out-of-capability data are the Medical Questions Pairs (MQP) (McCreery et al., 2020) differentiation (similar vs. different) and RTE (Dagan et al., 2006) natural language inference (entailment vs. not entailment) datasets.



(a) SVHN (in-distribution) (b) MNIST (near OOD) (c) CIFAR-10 (far OOD)

Figure 5: Generative fill completions on the SVHN, MNIST, and CIFAR-10 test sets. Completions are sensible for the in-distribution task, but contain artifacts and hallucinations for the OOD tasks.

For imaging ICL experiments, we use SVHN for in-distribution data (Netzer et al., 2011), MNIST as "near" OOD data (LeCun et al., 1998), and CIFAR-10 as "far" OOD data (Krizhevsky et al., 2009). Our Llama-2 regression model takes a sequence of flattened, grayscale, 8x8 images as input. It is fit to random sequences of 16 images from the SVHN "extra" split, which has over 500k examples. A series of in-distribution generative fill tasks is shown in Figure 5a. In each row, the model is prompted with three in-context examples and asked to complete the missing half of the 4th example. Each completion in the "fill" column is sensible, even when the completed number differs from the "real" number. Figure 5b illustrates completions for near-OOD MNIST tasks. We see in rows 1, 2, 3, and 6 that the fills are often sensible, but the model is prone to hallucinating odd completions (row 4) and artifacts (row 5). Figure 5c illustrates completions for far-OOD CIFAR-10 tasks. The completions are surprisingly consistent at this resolution, but as the result in row 4 demonstrates, the model hallucinates completions from its domain.

Discrepancy functions. We evaluate the p-value using discrepancy functions defined as

$$g_{\boldsymbol{\theta}}(\mathrm{x}, \mathrm{x}^{(\cdot)}) := -\frac{1}{|\mathrm{x}|} \sum_{\mathrm{z}_i, \mathrm{y}_i \in \mathrm{x}} \frac{1}{|(\mathrm{z}_i, \mathrm{y}_i)|} \sum_{\mathrm{t}_j \in (\mathrm{z}_i, \mathrm{y}_i)} \log p_{\boldsymbol{\theta}}(\mathrm{t}_j \mid \mathrm{t}_{<j}, \mathrm{x}^{(\cdot)}),$$

where $|(\mathrm{z}_i, \mathrm{y}_i)|$ is the number of tokens in the evaluated example. Following this template, the per-token negative log marginal likelihood (NLML) is written $g_{\boldsymbol{\theta}}(\mathrm{x}, \mathrm{x}^n)$ and an estimate of the per-token the negative log-likelihood (NLL) is written $g_{\boldsymbol{\theta}}(\mathrm{x}, \mathrm{x}^N)$, where x^N is generated as in Algorithm 1.

Predicting model capability. The p-values are calculated using either Algorithm 1 or Algorithm 3. Selecting a significance level α yields a binary predictor of model capability $\mathbb{1}\{p_{\text{gpc}} < \alpha\}$; a model is predicted incapable if the estimated generative predictive p-value is less than the significance level. We report results for significance levels $\alpha \in [0.01, 0.05, 0.1, 0.2, 0.5]$. For the NLL discrepancy function, replication data x is independently sampled from the likelihood under a hypothetical dataset completion $p_{\boldsymbol{\theta}}(\mathrm{z}, \mathrm{y} \mid \mathrm{x}^N)$. For the NLML discrepancy function, replication data is independently sampled from the predictive distribution $p_{\boldsymbol{\theta}}(\mathrm{z}, \mathrm{y} \mid \mathrm{x}^n)$.

Evaluation metrics. We evaluate the capability predictor using standard metrics: FPR measures in-capability tasks misclassified as out-of-capability, Precision reflects correctly identified out-of-capability tasks, and Recall measures correctly detected out-of-capability tasks. F1 Score and Accuracy assess overall performance (see Figure 6 for definitions).

Figure 6: Evaluation metrics for GPC performance.

Metric	Equation
False Positive Rate (FPR)	$\frac{\text{False Positives}}{\text{False Positives}+\text{True Negatives}}$
Precision	$\frac{\text{True Positives}}{\text{True Positives}+\text{False Positives}}$
Recall	$\frac{\text{True Positives}}{\text{True Positives}+\text{False Negatives}}$
F1 Score	$\frac{2\cdot\text{Precision}\cdot\text{Recall}}{\text{Precision}+\text{Recall}}$
Accuracy	$\frac{\text{True Positives}+\text{True Negatives}}{\text{Total Number of Predictions}}$

We also provide the distribution of p-values across tasks to assess how confidently the model distinguishes between the different ICL problems. Lower p-values indicate stronger confidence that a model cannot solve a problem.

7.1 The generative predictive p-value accurately predicts model capability

Tabular data. We first evaluate whether the generative predictive p-value effectively predicts OOD tabular data tasks. The parameters for Algorithm 1 are $M = 40$ replications and $N - n = 200$ generated examples. The ICL dataset x^n size is varied from $n = 2$ to $n = 200$. Figure 7 plots precision, recall, F1, and accuracy curves and shows that the p-value estimates under either discrepancy function provide non-trivial OOD predictors for all α settings.

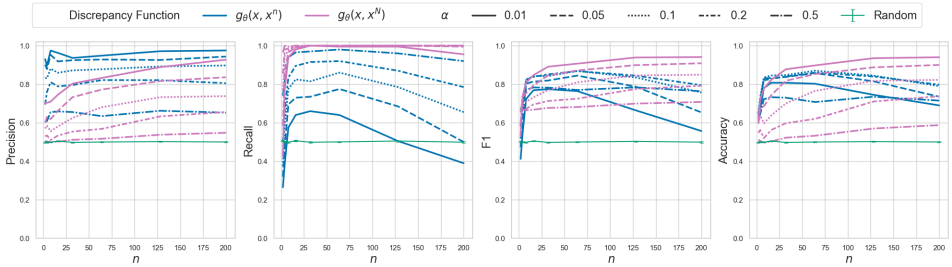


Figure 7: Tabular OOD detection. Metric values vs. context length. In-distribution functions are from unseen random ReLU-NNs. OOD functions are from an RBF kernel GP. Both discrepancy functions yield better than random OOD predictors for a range of α settings.

Natural language ICL. Next, we evaluate whether the generative predictive p-value effectively predicts out-of-capacity natural language tasks. The parameters for Algorithm 1 are $M = 20$ replications and $N - n = 10$ generated examples. The ICL dataset x^n size is varied from $n = 4$ to $n = 64$. Figure 8 plots precision, recall, F1, and accuracy curves and shows that the p-value estimates under the NLL discrepancy provide non-trivial (accuracy > 0.5) out-of-capability predictors in the domain of natural language for all α settings. The NLML discrepancy $g_{\boldsymbol{\theta}}(\mathrm{x}, \mathrm{x}^n)$ is also generally robust outside of the small n and small α setting.

Generative fill. Finally, we evaluate whether the generative predictive p-value effectively predicts OOD generative fill tasks. The parameters for Algorithm 1 are $M = 100$ replications and $N - n = 8$ generated examples The ICL dataset x^n size is varied from $n = 2$ to $n = 8$. Figure 9 plots the

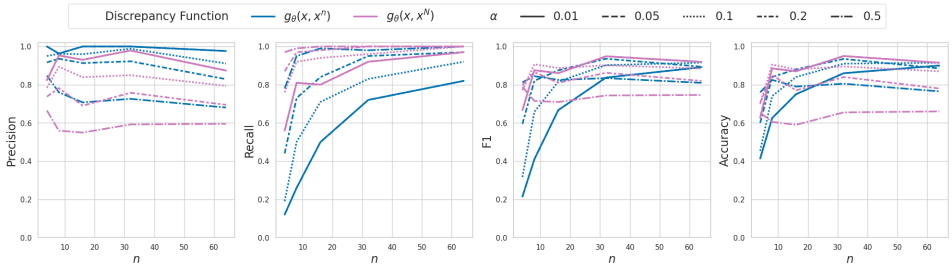


Figure 8: Llama-2-7B out-of-capability detection. Metric values vs. context length. In-capability tasks are from SST2 and AG News datasets. Out-of-capability tasks are from RTE and MQP datasets. Both discrepancy functions yield better than random OOD predictors for a range of α settings.

OOD prediction metric curves and shows that the p-value estimates under either discrepancy function provide non-trivial (accuracy $> 0.66\overline{6}$) OOD predictors for all α settings.

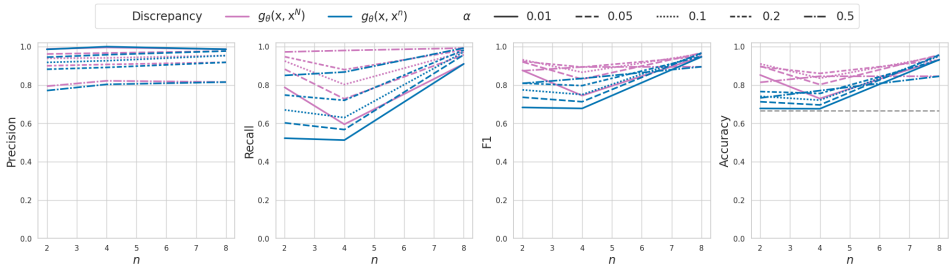
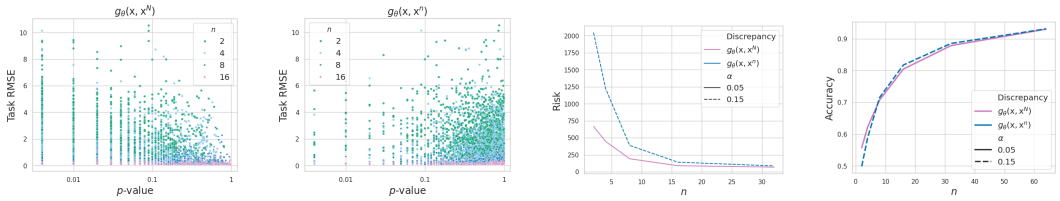


Figure 9: Generative fill OOD detection. Metric values vs. context length. In-distribution tasks are from the SVHN test set. Near and far OOD tasks are from the MNIST and CIFAR-10 test sets. Both discrepancy functions yield better than random OOD predictors for a range of α settings.

Discussion. Figures 7 to 9 reveal several trends. First, the NLML discrepancy (blue) yields better precision, indicating that it is less likely to misclassify an in-capability ICL problem as unsolvable. Second, the NLL discrepancy (purple) yields higher recall, indicating that it is less likely to misclassify an out-of-capability ICL problem as solvable. Third, the NLL discrepancy with significance level $\alpha = 0.05$ yields a generally more accurate predictor than the NLML discrepancy function for any significance level in the set evaluated. Finally, the recall of a predictor under the NLML discrepancy is sensitive to the number of in-context examples n. Next, we look deeper into the relationship between dataset size and the discrepancy functions.

7.2 The NLL discrepancy also indicates whether you have enough data



(a) NLL (b) NLML (c) Risk (d) Accuracy

Figure 10: With equal accuracy, the NLML discrepancy can yield a classifier with lower risk. Figures 10a and 10b are scatter plots of response RMSE vs. p-values for each discrepancy function. Points are coded by ICL dataset size n. Figure 10c Risk vs. n. Figure 10d Accuracy vs. n.

Both discrepancy functions yield accurate predictors of model capability, but the NLL discrepancy tells us if there are enough in-context examples to reliably solve a task. We use prediction RMSE

over task responses to measure reliability. Figures 10a and 10b plot the RMSE against the p-values computed under the NLL and NLML discrepancies for in-distribution polynomial tabular tasks. We see that lower p-values correlate with higher RMSE for the NLL discrepancy, but not for the NLML discrepancy. This information is useful for reducing risk in autonomous recommendation systems that respond if the p-value is greater than the significance level α. For example, at $\alpha = 0.1$, the NLL discrepancy reduces the generation of responses with high error because it accounts for the number of examples provided. Taking the risk as the sum of task RMSEs for tasks predicted as in-capability, Figures 10c and 10d show that the NLL discrepancy results in substantially reduced risk, even when the accuracies of each predictor are matched. Figure 14 in the appendix gives further insight into how the distributions of p-values evolve with dataset size for each discrepancy function.

7.3 The number of generated examples $N - n$ interpolates the p-value estimate between the NLML and the ideal NLL discrepancies

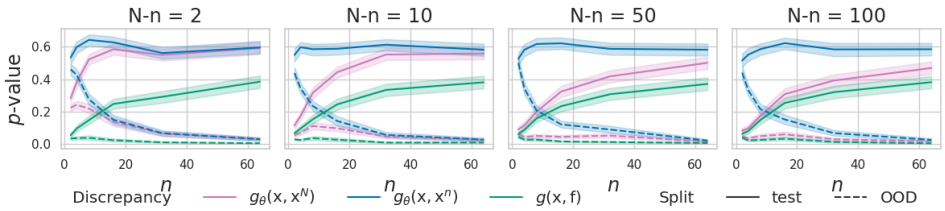


Figure 11: The p-value is interpolated between the NLL estimate and the NLML estimate as completion size $N - n$ grows from 0 to ∞.

Inspection of Equations (1) to (3) shows that the dataset completion size $N - n$ should interpolate p-value estimates between p_{ppc} computed with the NLML discrepancy and with the NLL discrepancy using the likelihood and posterior of a Bayesian model. To verify this, we use a reference Bayesian polynomial regression model to compute the p_{ppc}. We use our Llama-2 regression model fit to datasets generated from the reference model likelihood under different explanations to compute the p_{gpc}. We let datasets generated by random ReLU-NNs serve as OOD tasks. Figure 11 demonstrates that our expectation is true. Specifically, the p-value estimates at $N - n = 2$ are distributionally close to those calculated under the NLML, and they more closely approximate those calculated under the reference NLL discrepancy as we increase $N - n$ to 100. The latter observation is also illustrated in Figure 13.

Since the p-values from either discrepancy accurately predict model capability, the choice depends on whether the added computational cost of generating dataset completions is justified. If you need to know whether there are enough in-context examples to generate an accurate response—a necessity in risk-sensitive applications—then we recommend using the NLL discrepancy function. If computational efficiency or the cost of response deferral are primary concerns—practical user experience concerns—we suggest using the NLML discrepancy.

8 Conclusion

This work introduces the *generative predictive p-value*, a metric for determining whether a Conditional Generative Model can solve an in-context learning problem. It extends Bayesian model criticism techniques like PPCs to generative models by sampling dataset completions from the model's predictive distribution to approximate sampling latent explanations from a Bayesian model posterior. Empirical evaluations on tabular, natural language, and imaging tasks show that the generative predictive p-value can effectively identify the limits of model capability, distinguishing between in-capability and out-of-capability tasks for models like Llama-2 7B and Gemma-2 9B. This approach is a practical method to assess model suitability that advances Bayesian model criticism for CGMs. While we have focused on model capability prediction, the p-value estimates could also be used for model selection or as a general measure of task uncertainty. We are eager to explore extensions beyond ICL tasks to improve the reliability of generative AI systems.

9 ACKNOWLEDGMENTS

This work is supported by the funds provided by the National Science Foundation and by DoD OUSD (R&E) under Cooperative Agreement PHY-2229929 (The NSF AI Institute for Artificial and Natural Intelligence). Additional support was provided by the National Science Foundation under grants IIS-2127869 and DMS-2311108, the Office of Naval Research under grant N000142412243, and the Simons Foundation.

REFERENCES

Amos Azaria and Tom Mitchell. The internal state of an LLM knows when it's lying. In *EMNLP*, pp. 967–976, 2023.

Neil Band, Xuechen Li, Tengyu Ma, and Tatsunori Hashimoto. Linguistic calibration of language models. *arXiv:2404.00474*, 2024.

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

Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. Discovering latent knowledge in language models without supervision. In *ICLR*, 2023.

Jiuhai Chen and Jonas Mueller. Quantifying uncertainty in answers from any language model and enhancing their trustworthiness. In *ACL*, pp. 5186–5200, 2024.

Yung-Sung Chuang, Yujia Xie, Hongyin Luo, Yoon Kim, James R Glass, and Pengcheng He. Dola: Decoding by contrasting layers improves factuality in large language models. In *ICLR*, 2024.

Jeremy R Cole, Michael JQ Zhang, Daniel Gillick, Julian Martin Eisenschlos, Bhuwan Dhingra, and Jacob Eisenstein. Selectively answering ambiguous questions. *EMNLP*, 2023.

Ido Dagan, Oren Glickman, and Bernardo Magnini. The pascal recognising textual entailment challenge. In Joaquin Quiñonero-Candela, Ido Dagan, Bernardo Magnini, and Florence d'Alché Buc (eds.), *Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Tectual Entailment*, pp. 177–190, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg. ISBN 978-3-540-33428-6.

Bruno de Finetti. La prévision: ses lois logiques, ses sources subjectives. *Annales de l'Institut Henri Poincaré*, 7:1–68, 1937.

Shehzaad Dhuliawala, Mojtaba Komeili, Jing Xu, Roberta Raileanu, Xian Li, Asli Celikyilmaz, and Jason E Weston. Chain-of-verification reduces hallucination in large language models. In *ICLR 2024 Workshop on Reliable and Responsible Foundation Models*, 2024.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, and Zhifang Sui. A survey on in-context learning. *arXiv preprint arXiv:2301.00234*, 2022.

Joseph L Doob. Application of the theory of martingales. *Le calcul des probabilites et ses applications*, pp. 23–27, 1949.

Nouha Dziri, Andrea Madotto, Osmar R Zaiane, and Avishek Joey Bose. Neural path hunter: Reducing hallucination in dialogue systems via path grounding. In *EMNLP*, pp. 2197–2214, 2021.

Mohamed Elaraby, Mengyin Lu, Jacob Dunn, Xueying Zhang, Yu Wang, and Shizhu Liu. Halo: Estimation and reduction of hallucinations in open-source weak large language models. *arXiv:2308.11764*, 2023.

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

Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630(8017):625–630, 2024.

Philip Feldman, James R Foulds, and Shimei Pan. Trapping LLM hallucinations using tagged context prompts. *arXiv:2306.06085*, 2023.

Edwin Fong, Chris Holmes, and Stephen G Walker. Martingale posterior distributions. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 85(5):1357–1391, 2023.

Luyu Gao, Zhuyun Dai, Panupong Pasupat, Anthony Chen, Arun Tejasvi Chaganty, Yicheng Fan, Vincent Zhao, Ni Lao, Hongrae Lee, Da-Cheng Juan, et al. RARR: Researching and revising what language models say, using language models. In *ACL*, pp. 16477–16508, 2023.

Andrew Gelman, Xiao-Li Meng, and Hal Stern. Posterior predictive assessment of model fitness via realized discrepancies. *Statistica sinica*, pp. 733–760, 1996.

Ishaan Gulrajani and Tatsunori B Hashimoto. Likelihood-based diffusion language models. *Advances in Neural Information Processing Systems*, 36:16693–16715, 2023.

Irwin Guttman. The use of the concept of a future observation in goodness-of-fit problems. *Journal of the Royal Statistical Society: Series B (Methodological)*, 29(1):83–100, 1967.

Edwin Shields Hewitt and Leonard J. Savage. Symmetric measures on cartesian products. *Transactions of the American Mathematical Society*, 80:470–501, 1955. URL https://api.semanticscholar.org/CorpusID:53585081.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Andrew Jesson, Nicolas Beltran-Velez, Quentin Chu, Sweta Karlekar, Jannik Kossen, Yarin Gal, John P Cunningham, and David Blei. Estimating the hallucination rate of generative ai. *arXiv e-prints*, pp. arXiv–2406, 2024.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield Dodds, Nova DasSarma, Eli Tran-Johnson, et al. Language models (mostly) know what they know. *arXiv:2207.05221*, 2022.

Jannik Kossen, Jiatong Han, Muhammed Razzak, Lisa Schut, Shreshth Malik, and Yarin Gal. Semantic entropy probes: Robust and cheap hallucination detection in llms. *arXiv preprint arXiv:2406.15927*, 2024.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. *University of Toronto*, 2009.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. *ICLR*, 2023.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Hyungi Lee, Eunggu Yun, Giung Nam, E Fong, and Juho Lee. Martingale posterior neural processes. In *The Eleventh International Conference on Learning Representations*. International Conference on Learning Representations, 2023.

Nayeon Lee, Wei Ping, Peng Xu, Mostofa Patwary, Pascale N Fung, Mohammad Shoeybi, and Bryan Catanzaro. Factuality enhanced language models for open-ended text generation. *NeurIPS*, 35: 34586–34599, 2022.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Inference-time intervention: Eliciting truthful answers from a language model. *NeurIPS*, 36, 2024.

Xingxuan Li, Ruochen Zhao, Yew Ken Chia, Bosheng Ding, Shafiq Joty, Soujanya Poria, and Lidong Bing. Chain-of-knowledge: Grounding large language models via dynamic knowledge adapting over heterogeneous sources. In *ICLR*, 2023.

Stephanie Lin, Jacob Hilton, and Owain Evans. Teaching models to express their uncertainty in words. *TMLR*, 2023.

Zhen Lin, Shubhendu Trivedi, and Jimeng Sun. Generating with confidence: Uncertainty quantification for black-box large language models. *TMLR*, 2024.

Junyu Luo, Cao Xiao, and Fenglong Ma. Zero-resource hallucination prevention for large language models. *CoRR*, 2023.

Potsawee Manakul, Adian Liusie, and Mark JF Gales. SelfCheckGPT: Zero-resource black-box hallucination detection for generative large language models. In *EMNLP*, 2023.

Clara H McCreery, Namit Katariya, Anitha Kannan, Manish Chablani, and Xavier Amatriain. Effective transfer learning for identifying similar questions: Matching user questions to covid-19 faqs. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 3458–3465, 2020.

Xiao-Li Meng. Posterior predictive p-values. *The annals of statistics*, 22(3):1142–1160, 1994.

Sabrina J Mielke, Arthur Szlam, Emily Dinan, and Y-Lan Boureau. Reducing conversational agents' overconfidence through linguistic calibration. *TACL*, 10:857–872, 2022.

Jeffrey W Miller. A detailed treatment of doob's theorem. *arXiv preprint arXiv:1801.03122*, 2018.

Gemma E Moran, David M Blei, and Rajesh Ranganath. Holdout predictive checks for [b]ayesian model criticism. In *2023 IMS International Conference on Statistics and Data Science (ICSDS)*, pp. 94, 2023.

Samuel Müller, Noah Hollmann, Sebastian Pineda Arango, Josif Grabocka, and Frank Hutter. Transformers can do bayesian inference. In *International Conference on Learning Representations*, 2021.

Niels Mündler, Jingxuan He, Slobodan Jenko, and Martin Vechev. Self-contradictory hallucinations of large language models: Evaluation, detection and mitigation. In *ICLR*, 2024.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Baolin Wu, Andrew Y Ng, et al. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*. Granada, 2011.

Harsha Nori, Yin Tat Lee, Sheng Zhang, Dean Carignan, Richard Edgar, Nicolò Fusi, Nicholas King, Jonathan Larson, Yuanzhi Li, Weishung Liu, et al. Can generalist foundation models outcompete special-purpose tuning? case study in medicine. *CoRR*, 2023.

Baolin Peng, Michel Galley, Pengcheng He, Hao Cheng, Yujia Xie, Yu Hu, Qiuyuan Huang, Lars Liden, Zhou Yu, Weizhu Chen, et al. Check your facts and try again: Improving large language models with external knowledge and automated feedback. *arXiv:2302.12813*, 2023.

Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Matt Turner. Steering Llama 2 via contrastive activation addition. *arXiv:2312.06681*, 2023.

Donald B Rubin. [b]ayesianly justifiable and relevant frequency calculations for the applied statistician. *The Annals of Statistics*, pp. 1151–1172, 1984.

Weijia Shi, Xiaochuang Han, Mike Lewis, Yulia Tsvetkov, Luke Zettlemoyer, and Wen-tau Yih. Trusting your evidence: Hallucinate less with context-aware decoding. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pp. 783–791, 2024.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *EMNLP*, pp. 1631–1642, 2013.

Dan Su, Xiaoguang Li, Jindi Zhang, Lifeng Shang, Xin Jiang, Qun Liu, and Pascale Fung. Read before generate! Faithful long form question answering with machine reading. In *ACL*, pp. 744–756, 2022.

Gemma Team. Gemma, 2024.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv:2307.09288*, 2023.

Neeraj Varshney, Wenlin Yao, Hongming Zhang, Jianshu Chen, and Dong Yu. A stitch in time saves nine: Detecting and mitigating hallucinations of LLMs by validating low-confidence generation. *arXiv:2307.03987*, 2023.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. An explanation of in-context learning as implicit bayesian inference. In *International Conference on Learning Representations*, 2021.

Naimeng Ye, Hanming Yang, Andrew Siah, and Hongseok Namkoong. Pre-training and in-context learning is bayesian inference a la de finetti. *arXiv preprint arXiv:2408.03307*, 2024.

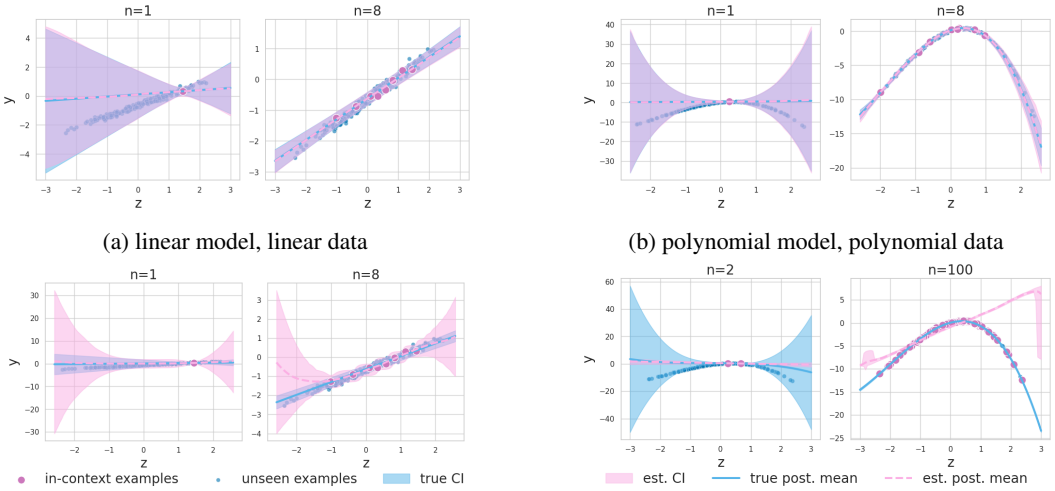
Shuo Zhang, Liangming Pan, Junzhou Zhao, and William Yang Wang. Mitigating language model hallucination with interactive question-knowledge alignment. *arXiv:2305.13669*, 2023.

Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. *NeurIPS*, 2015.

A MODEL INTUITION

In this section we give an interpretation of the component parts of a Bayesian model, how they are used to make inferences about uncertainty, and how to relate inferences in classical domains to inferences in more complex domains like language.

The model prior $p_{\boldsymbol{\theta}}(\mathrm{f})$ can be thought of as a library over the possible explanations a model could ascribe to observations. It is a special kind of library, where the probability of finding an explanation in the library at random is also defined. The model prior encodes everything "known" to a model; all the latent patterns available as explanations for—or an index of all the probability distributions ascribable to—any set of observations. The model prior may or may not assign non-zero probability to an explanation f equivalent to a given ICL problem task f^*. If no such explanation has coverage under the prior, then the model may not be able to provide an accurate solution to the ICL problem.



(a) linear model, linear data

(b) polynomial model, polynomial data

(c) polynomial model, linear data

(d) linear model, polynomial data

Figure 12: Examples of misaligned model and data combinations. Transformer models (pink) are fit to either linear or polynomial noisy data defined by reference Bayesian models (blue).

For example, a Bayesian linear regression model with fixed noise, defines the part of the prior over explanations $p_{\boldsymbol{\theta}}(\mathrm{f})$ related to the outcome y as a set of coefficient vectors; a set of hyperplanes. If the ICL dataset and prediction task are characterized by a linear relationship between continuous valued queries and responses—illustrated in Figure 12a—the prior would be suitable for the ICL problem. However, if they are characterized by polynomial relationships—illustrated in Figure 12c—then the relationship would not have coverage under the prior and the precision of responses under the model would be limited by the capacity of a hyperplane to fit a polynomial surface. By analogy, a LLM that is pre-trained or fine-tuned on a large set of integrals expressed in natural language may have the functional capacity to integrate; generalize to unseen functions and domains of the classes and spaces covered in the training set. So if the ICL dataset and task are related to the integration of polynomials, the learned library of mappings from text to token distributions may be appropriate for that ICL problem. However, if the LLM training corpus did not contain content related to calculus, then learned library of mappings may not include the functional capability to solve the ICL problem.

The model likelihood $p_{\boldsymbol{\theta}}(\mathrm{x} \mid \mathrm{f})$ encodes the variety of observations x that could be generated according to a given explanation f. The variety encoded by this distribution is often called *aleatoric* uncertainty—aleatory is a pretentious word for random—which refers to the inherent *randomness* over generated datasets when sampling according to the likelihood under a given explanation f. For example, given the explanation implied by a fair coin, we will still be *uncertain* whether the outcome of a single coin flip will be heads or tails. More contemporarily, if you are already familiar with this concept and I were to say, "I would like to share the idea of aleatoric uncertainty with you," you would know *which* idea I want to share, but, before reading this paragraph, you would be uncertain about *how* I would share it with you. When the model likelihood $p_{\boldsymbol{\theta}}(\mathrm{x} \mid \mathrm{f})$ is indexed by an explanation

that is equivalent to the task $\mathrm{f} \equiv \mathrm{f}^*$, then $p_{\boldsymbol{\theta}}(\mathrm{x} \mid \mathrm{f})$ is equal to the reference likelihood $p(\mathrm{x} \mid \mathrm{f}^*)$. So even though we may be uncertain about *which* dataset would be generated according to the model likelihood, we could still be certain that the generated dataset would correspond to the task. However, if there is a discrepancy between the model and reference likelihood then the model may not be suitable for the in-context learning problem.

The model posterior

$$p_{\boldsymbol{\theta}}(\mathrm{f} \mid \mathrm{x}^n) = \frac{p_{\boldsymbol{\theta}}(\mathrm{x}^n \mid \mathrm{f}) p_{\boldsymbol{\theta}}(\mathrm{f})}{\int p_{\boldsymbol{\theta}}(\mathrm{x}^n \mid \mathrm{f}) d\mathrm{P}_{\boldsymbol{\theta}}(\mathrm{f})},$$

encodes variety over explanations that *could* have generated a specific set of observations x^n. This variety is often called *epistemic* uncertainty. Epistemic is a pretentious term referring to knowledge, conveying that we may not yet know *which* explanation f among subset of reasonable explanations best explains possible datasets x under the ICL problem.

For example, given only four observations—say, two heads and two tails—the sample mean estimate for the probability of observing heads is 0.5. However, we may still be uncertain about whether the coin generating the outcomes was fair or biased because the variance of that estimate is still a non-negligible 0.0625 when we assume the coin is actually fair. Related to our contemporary example, if I only say, "I would like to share an idea with you," you can probably imagine an abundance of ideas that I could be referring to and thus still be uncertain about which one I have chosen to share.

A relevant feature of epistemic uncertainty is that it is reducible as we observe more context. In the coin flip example, as we observe more outcomes, our certainty about the probability of heads increases. In the second example, you may have a better idea about the class of ideas I may share based on what has been presented thus far.

As a function of both the model likelihood and prior, the model posterior inherits the limitations of both. But it also provides information about whether an in-context learning problem can be solved reliably. Namely, variety over explanations is indicative of being uncertain about which task the ICL dataset corresponds to. This variety can lead to the model generating responses corresponding to tasks other than f^*. But it may also be indicative of when more examples (larger n) can improve the quality of solutions to an in-context learning problem.

The model posterior predictive

$$p_{\boldsymbol{\theta}}(\mathrm{x} \mid \mathrm{x}^n) = \int p_{\boldsymbol{\theta}}(\mathrm{x} \mid \mathrm{f}) d\mathrm{P}_{\boldsymbol{\theta}}(\mathrm{f} \mid \mathrm{x}^n),$$

is derived from the model to generate new observations x given past observations x^n. Poetically, the model posterior predictive gives the model a voice to respond to observations with observations. The model posterior predictive convolves the model likelihood of the observations given an explanation with the model posterior over explanations. This process entangles variety over explanations after observing a dataset x^n and variety over observations x for each specific explanation f; the model posterior predictive entangles aleatoric and epistemic uncertainty.

Model inferences. A model $\boldsymbol{\theta}$ defined over an observation space $\mathcal{X}$ is used make inferences about observations from that space x^n. Inferences like the probability of the the next word given a sequence of words, model uncertainty, and countless other things. But a model is a choice—the practitioner makes a modeling decision—and so the inferences derived from observations under a model may or may not be grounded in reality.

Figure 12 illustrates inferences about the predictive distribution made by different models given different datasets. When the data and model are well aligned— Figures 2a and 12b—model inferences (pink) overlap with those made by the reference model (blue) and appear purple. However, inferences made by a misaligned polynomial model—Figure 12c—are much wider that those made by the reference linear model. And inferences made by a misaligned linear model—Figure 12d—are more narrow than those ogf the reference polynomial model, which results in the model being confident and wrong. Similarly, a very general LLM may have compromised data efficiency for rare domains, while a highly specialized LLM may no longer generalize beyond its domain.

These examples illustrate that while models are used to quantify empirical facts—like the frequency of an event occurring—they also carry a subjective aspect that needs to be considered when using model inferences in practice. This consideration guides our question of when a model will provide reliable inferences for a given ICL problem.

B Defining model capability for an in-context learning problem

In this section, we formalize the idea that, "a model *can* solve an ICL problem." A formal definition must account for two things: (1) the model may generate undesired responses because the context *does not* adequately specify the task, and (2) the model may generate undesired responses *despite* the context precisely specifying the task. The definition below accounts for both.

Definition 1. *An ICL problem comprises a model $\boldsymbol{\theta}$, a dataset $\mathrm{x}^n = \{\mathrm{z}_i, \mathrm{y}_i\}_{i=1}^n \sim p(\mathrm{x}^n \mid \mathrm{f}^*)$, and a task f^*. Assume that valid responses y to user queries $\mathrm{z} \sim p(\mathrm{z} \mid \mathrm{f}^*)$ are distributed as $p(\mathrm{y} \mid \mathrm{z}, \mathrm{f}^*)$ under the task. Finally, let $A(\mathrm{z}, \mathrm{f}^*)$ denote any set of responses satisfying*

$$Pr(\mathrm{Y} \in A(\mathrm{z}, \mathrm{f}^*) \mid \mathrm{z}, \mathrm{f}^*) \geq 1 - \epsilon. \tag{4}$$

The model $\boldsymbol{\theta}$ is called capable of solving the ICL problem if

$$\lim_{n \to \infty} \int \mathbb{1}\left\{\mathrm{y} \in A(\mathrm{z}, \mathrm{f}^*)\right\} d\mathrm{P}_{\boldsymbol{\theta}}(\mathrm{y} \mid \mathrm{z}, \mathrm{x}^n) \geq 1 - \epsilon. \tag{5}$$

Equation (4) defines a set of valid responses y to any query z given the task f^*. Jesson et al. (2024) call this a $(1 - \epsilon)$-likely set. For example, the set could be a confidence interval in a regression task or a set of semantically equivalent ways to express positive sentiment in an open-ended sentiment analysis task. The $1 - \epsilon$ set gives us a formal and general way to express the notion of a desirable response for a given query and task. Equation (5) then says that a model is capable if the probability that a generated response belongs to the set of valid responses converges to be at least $1 - \epsilon$ as the size of the dataset grows; as the context more precisely specifies the task. This definition accounts for condition (1) through the limit, allowing for capable models with too few in-context examples to be called capable. The indicator accounts for both conditions by counting the number of times model-generated responses fall inside the $1 - \epsilon$ set; a general measure of accuracy.

In addition to accounting for the above conditions, this definition allows the model predictive distribution to collapse to subsets of $A(\mathrm{z}, \mathrm{f}^*)$—even deterministic responses—and still be called capable. This attribute is preferable to a definition of capability that requires the model predictive distribution to converge to the reference distribution, which would exclude many practical models.

While this definition is general, there are still practical limitations to consider. For example, an infinitely deep and wide random transformer with a finite maximum sequence length might be capable of solving most problems under this definition; however, its data efficiency may be so poor that we fill the context window before it can generate accurate responses. Similarly, even if the model could accommodate infinitely long sequences, the available data may exhausted before the model generates desirable responses. These scenarios are extreme examples of the over-parameterized case illustrated in Figure 2c.

This discussion also sheds light on the propensity for a classifier using the NLML p-value to produce false negatives (misclassify out-of-capability tasks as in-capability). Focusing on the $\mathrm{z} > 0.5$ region of Figure 3b, the model will be predicted as in-capability because the model predictive distribution covers the data for small in-context learning dataset sizes. It is not until the model sees 100 examples that the misalignment between the predicted and reference distributions becomes apparent. The NLL discrepancy addresses this failure mode, as discussed.

C Proofs for theoretical results

We restate our formalization and assumptions for convenience. Observable examples $(\mathrm{z}, \mathrm{y}) \in \mathcal{X}$ are modeled by the $(\mathcal{X}, \mathcal{A})$-random variable X_i and explanations $\mathrm{f} \in \mathcal{F}$ are modeled by the $(\mathcal{F}, \mathcal{B})$-random variable F, where $\mathcal{A}$ and $\mathcal{B}$ are the relevant sigma algebras. For each $\mathrm{f} \in \mathcal{F}$, let the model $\boldsymbol{\theta}$ define a probability measure $\mathrm{P}_{\boldsymbol{\theta}}^{\mathrm{f}}$ on $(\mathcal{X}, \mathcal{A})$. Let the model $\boldsymbol{\theta}$ further define a probability measure $\mathrm{P}_{\boldsymbol{\theta}}$ on $(\mathcal{F}, \mathcal{B})$. And let $\mathrm{P}_{\boldsymbol{\theta}}$ and $\mathrm{P}_{\boldsymbol{\theta}}^{\mathrm{f}}$ define the joint measure $\mathrm{M}_{\boldsymbol{\theta}}$ over $((\mathrm{X}_1, \mathrm{X}_2, \dots), \mathrm{F})$. Finally, we overload the notation "$\sim$." It means "sampled according to" when referring to the relationship between a random variable instance and a density or distribution; e.g., $\mathrm{x} \sim p_{\boldsymbol{\theta}}(\mathrm{x})$. And it means "distributed as" when referring to the relationship between a random variable and a probability measure; e.g., $\mathrm{X} \sim \mathrm{P}_{\boldsymbol{\theta}}^{\mathrm{f}}$. Our method rests on Doob's theorem for estimators (Doob, 1949), which assumes the following three conditions.

Condition 1. *The observation $\mathcal{X}$ and explanation $\mathcal{F}$ spaces are complete and separable metric spaces.*

Condition 2. *The set of probability measures $\{\mathrm{P}^{\mathrm{f}}_{\boldsymbol{\theta}} : \mathrm{f} \in \mathcal{F}\}$ defined by the model θ is a measurable family; the mapping $\mathrm{f} \mapsto \mathrm{P}^{\mathrm{f}}_{\boldsymbol{\theta}}(A)$ is measurable for every $A \in \mathcal{A}$.*

Condition 3. *The model $\boldsymbol{\theta}$ is identifiable;*

$$\mathrm{f} \neq \mathrm{f}' \Rightarrow \mathrm{P}^{\mathrm{f}}_{\boldsymbol{\theta}} \neq \mathrm{P}^{\mathrm{f}'}_{\boldsymbol{\theta}}. \tag{6}$$

Given these conditions we can state Doob's theorem.

Theorem 2. ***Doob's Theorem for estimators.*** *Let $\mathrm{F} \sim \mathrm{P}_{\boldsymbol{\theta}}$ and $\mathrm{X}_1, \mathrm{X}_2, \ldots$ i.i.d $\sim \mathrm{P}^{\mathrm{f}}_{\boldsymbol{\theta}}$. Assume Conditions 1 to 3 and a measurable function $h : \mathcal{F} \to \mathbb{R}$ such that $\int |h(\mathrm{f})| d\mathrm{P}_{\boldsymbol{\theta}}(\mathrm{f}) < \infty$, then*

$$\lim_{n\to\infty} \mathbb{E}[h(\mathrm{F}) \mid \mathrm{X}_1, \mathrm{X}_2, \ldots, \mathrm{X}_n] = h(\mathrm{F}) \text{ a.s. } [\mathrm{M}_{\boldsymbol{\theta}}]. \tag{7}$$

Proof. Miller (2018) provides a detailed proof of this theorem. □

Lemma 3. *Let $\mathrm{F} \sim \mathrm{P}_{\boldsymbol{\theta}}$, and $\mathrm{X}_1, \mathrm{X}_2, \ldots$ i.i.d $\sim \mathrm{P}^{\mathrm{f}}_{\boldsymbol{\theta}}$. Assume Conditions 1 to 3 and let $\{\int |\log p_{\boldsymbol{\theta}}(\mathrm{x}^m \mid \mathrm{f})| d\mathrm{P}_{\boldsymbol{\theta}}(\mathrm{f}) < \infty : \forall \mathrm{x}^m \in \mathcal{X}^m\}$. Then,*

$$g_{\boldsymbol{\theta}}(\mathrm{x}, \mathrm{F}) = -\frac{1}{|\mathrm{x}|} \log p_{\boldsymbol{\theta}}(\mathrm{x} \mid \mathrm{F}) = -\frac{1}{|\mathrm{x}|} \log p_{\boldsymbol{\theta}}(\mathrm{x} \mid \mathrm{X}^\infty) = g_{\boldsymbol{\theta}}(\mathrm{x}, \mathrm{X}^\infty).$$

Proof.

$$\begin{aligned}
p_{\boldsymbol{\theta}}(\mathrm{x} \mid \mathrm{F}) &= \lim_{n\to\infty} \int p_{\boldsymbol{\theta}}(\mathrm{x} \mid \mathrm{f}) d\mathrm{P}_{\boldsymbol{\theta}}(\mathrm{f} \mid \mathrm{X}_1, \ldots, \mathrm{X}_n) \\
&= \lim_{n\to\infty} \int p_{\boldsymbol{\theta}}(\mathrm{x} \mid \mathrm{f}, \mathrm{X}_1, \ldots, \mathrm{X}_n) d\mathrm{P}_{\boldsymbol{\theta}}(\mathrm{f} \mid \mathrm{X}_1, \ldots, \mathrm{X}_n) \\
&= \lim_{n\to\infty} p_{\boldsymbol{\theta}}(\mathrm{x} \mid \mathrm{X}_1, \ldots, \mathrm{X}_n)
\end{aligned}$$

$$\begin{aligned}
g(\mathrm{x}, \mathrm{F}) &= -\frac{1}{|\mathrm{x}|} \log p_{\boldsymbol{\theta}}(\mathrm{x} \mid \mathrm{F}) \\
&= -\frac{1}{|\mathrm{x}|} \log \lim_{n\to\infty} p_{\boldsymbol{\theta}}(\mathrm{x} \mid \mathrm{X}_1, \ldots, \mathrm{X}_n) \\
&= \lim_{n\to\infty} -\frac{1}{|\mathrm{x}|} \log p_{\boldsymbol{\theta}}(\mathrm{x} \mid \mathrm{X}_1, \ldots, \mathrm{X}_n) \\
&= -\frac{1}{|\mathrm{x}|} \log p_{\boldsymbol{\theta}}(\mathrm{x} \mid \mathrm{X}^\infty) \\
&= g(\mathrm{x}, \mathrm{X}^\infty)
\end{aligned}$$

□

Theorem 2. *Under the conditions of Lemma 3,*

$$p_{ppc} = p_{mpc}$$

Proof. Define an alternative probability model such that $\mathrm{F} \sim \mathrm{P}^{\mathrm{x}^n}_{\boldsymbol{\theta}}$ and $\mathrm{X}_1, \mathrm{X}_2, \ldots$ i.i.d $\sim \mathrm{P}^{\mathrm{f},\mathrm{x}^n}_{\boldsymbol{\theta}}$. Let p_a, P_b, and g_a denote the relevant quantities respecting this model. For example, $p_a(\mathrm{y} \mid \mathrm{x}) = p_{\boldsymbol{\theta}}(\mathrm{y} \mid \mathrm{x}, \mathrm{x}^n)$ and $P_a(\mathrm{f}) = \mathrm{P}_{\boldsymbol{\theta}}(\mathrm{f} \mid \mathrm{x}^n)$. Note that $p_{\boldsymbol{\theta}}(\mathrm{x} \mid \mathrm{f}) = p_{\boldsymbol{\theta}}(\mathrm{x} \mid \mathrm{f}, \mathrm{x}^n) = p_a(\mathrm{x} \mid \mathrm{f})$ since X and X^n are

independent when f is known.

$$
\begin{aligned}
p_{\text{ppc}} &= \iint \mathbb{1}\left\{g_{\boldsymbol{\theta}}(\mathrm{x},\mathrm{f}) \geq g_{\boldsymbol{\theta}}(\mathrm{x}^{\text{test}},\mathrm{f})\right\} d\mathrm{P}_{\boldsymbol{\theta}}(\mathrm{x} \mid \mathrm{f}) d\mathrm{P}_{\boldsymbol{\theta}}(\mathrm{f} \mid \mathrm{x}^n) \\
&= \int \mathbb{1}\left\{g_a(\mathrm{x},\mathrm{f}) \geq g_a(\mathrm{x}^{\text{test}},\mathrm{f})\right\} dP_a(\mathrm{x},\mathrm{f}) \\
&= \int \mathbb{1}\left\{g_a(\mathrm{x},\mathrm{f}) \geq g_a(\mathrm{x}^{\text{test}},\mathrm{f})\right\} dP_a(\mathrm{x},\mathrm{f},\mathrm{x}^{n+1:\infty}) \\
&= \iint \mathbb{1}\left\{g_a(\mathrm{x},\mathrm{f}) \geq g_a(\mathrm{x}^{\text{test}},\mathrm{f})\right\} dP_a(\mathrm{x},\mid \mathrm{f},\mathrm{x}^{n+1:\infty}) dP_a(\mathrm{f},\mathrm{x}^{n+1:\infty}) \\
&= \iint \mathbb{1}\left\{g_a(\mathrm{x},\mathrm{x}^{n+1:\infty}) \geq g_a(\mathrm{x}^{\text{test}},\mathrm{x}^{n+1:\infty})\right\} dP_a(\mathrm{x} \mid \mathrm{f},\mathrm{x}^{n+1:\infty}) dP_a(\mathrm{f},\mathrm{x}^{n+1:\infty}) \quad \text{Lemma 3} \\
&= \iint \mathbb{1}\left\{g_a(\mathrm{x},\mathrm{x}^{n+1:\infty}) \geq g_a(\mathrm{x}^{\text{test}},\mathrm{x}^{n+1:\infty})\right\} dP_a(\mathrm{x} \mid \mathrm{x}^{n+1:\infty}) dP_a(\mathrm{x}^{n+1:\infty}) \\
&= \iint \mathbb{1}\left\{g_{\boldsymbol{\theta}}(\mathrm{x},\mathrm{x}^{\infty}) \geq g_{\boldsymbol{\theta}}(\mathrm{x}^{\text{test}},\mathrm{x}^{\infty})\right\} d\mathrm{P}_{\boldsymbol{\theta}}(\mathrm{x} \mid \mathrm{x}^{\infty}) d\mathrm{P}_{\boldsymbol{\theta}}(\mathrm{x}^{n+1:\infty} \mid \mathrm{x}^n) \\
&= p_{\text{mpc}}
\end{aligned}
$$

□

D POSTERIOR PREDICTIVE P-VALUE ALGORITHM

Algorithm 2 details the procedure for calculating the posterior predictive p-value in Equation (1) given train data x^n, test data x^{test}, a discrepancy function $g_{\boldsymbol{\theta}}(\mathrm{x},\mathrm{f})$, and the nuber of replication datasets to generate M.

Algorithm 2 $\widehat{p}_{\text{ppc}}$

Require: data $\{\mathrm{x}^n, \mathrm{x}^{\text{test}}\}$, discrepancy function $g_{\boldsymbol{\theta}}(\mathrm{x},\mathrm{f})$, # replicates M

```
for i ← 1 to M do
    f_i ~ p_θ(f | x^n)                          ▷ sample explanation f
    x_i ← ()                                    ▷ initialize replicant data
    for j ← 1 to n do
        z_j, y_j ~ p_θ(z, y|f_i)                ▷ sample example from model likelihood
        x_i ← (x_i, z_j, y_j)                   ▷ update replicant data
return (1/M) Σ_{i=1}^{M} 𝟙{g_θ(x_i, f_i) ≥ g_θ(x^test, f_i)}   ▷ estimate p-value
```

E LITE GENERATIVE PREDICTIVE P-VALUE ALGORITHM

Algorithm 3 $\widehat{p}^{\text{lite}}_{\text{gpc}}$

Require: data $\{\mathrm{x}^n, \mathrm{x}^{\text{test}}\}$, a discrepancy function $g_{\boldsymbol{\theta}}(\mathrm{x},\mathrm{x}^n)$, # replicates M

```
for i ← 1 to M do
    x_i ← ()                                    ▷ initialize replicant data
    for j ← 1 to n do
        z_j, y_j ~ p_θ(z, y | x_i, x^n)         ▷ sample example from model
        x_i ← (x_i, z_j, y_j)                   ▷ update replicant data
return (1/M) Σ_{i=1}^{M} 𝟙{g_θ(x_i, x^n) ≥ g_θ(x^test, x^n)}   ▷ estimate p-value
```

Algorithm 3 summarizes a "lite" version of the estimator that forgoes approximate sampling from the model posterior and likelihood. Instead, it samples replication data directly from the model predictive distribution and calculates the discrepancy functions with respect to the observed data x^n rather than a dataset completion x^N.

F ADDITIONAL FIGURES

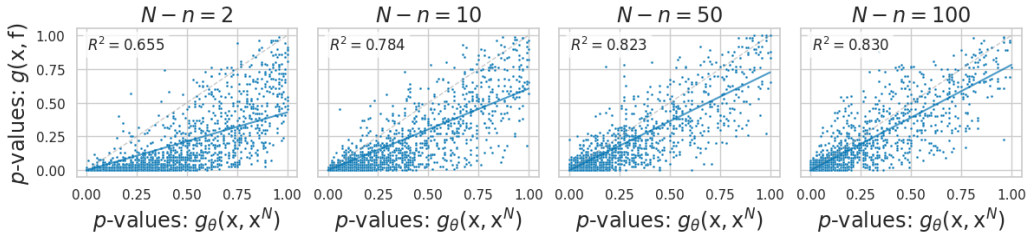
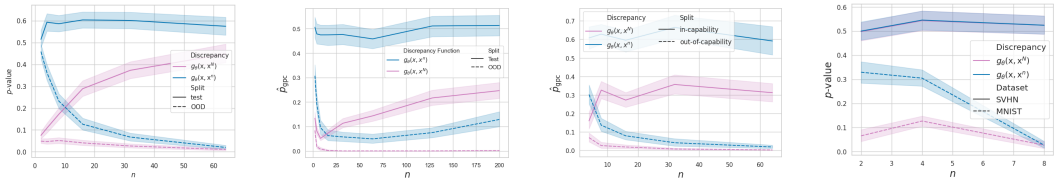


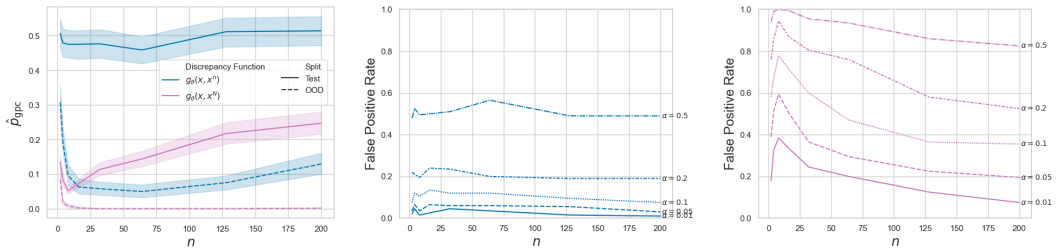
Figure 13: Scatter plots demonstrating that p_{gpc} becomes a better approximation of p_{ppc} with increasing dataset completion size $N - n$.



(a) Polynomial Tabular (b) ReLU-NN Tabluar (c) Natural language (d) Generative fill near

Figure 14: The generative predictive p-value against dataset size n

Figure 15a shows p-values as a function of the ICL dataset x^n size n (context length). We see that there is clear separation between the estimated generative predictive p-values $\widehat{p}_{\text{gpc}}$ for the in-distribution test set (solid lines) and the OOD dataset (dashed lines). The separation is robust across different ICL dataset sizes and the two discrepancy functions we test.



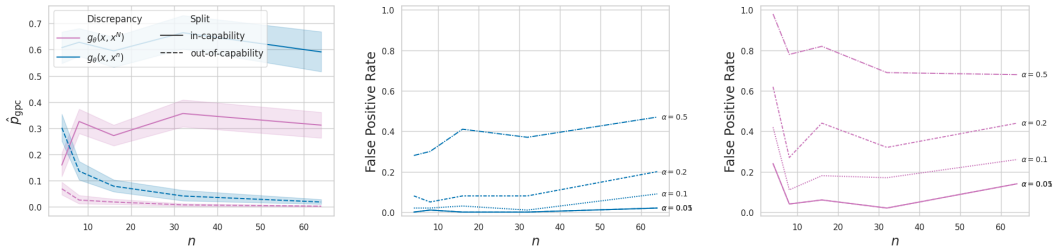
(a) Generative predictive p-values (b) False Positive Rate (NLML) (c) False Positive Rate (NLL)

Figure 15: Simulated regression task. The generative predictive p-value against dataset size n and it's relationship to the false positive rate. The first figure shows the generative predictive p-values, and the second and third figures show the false positive rate with the NLML and NLL discrepancy functions, respectively.

Figure 15b plots the FPR for the capability detector defined by $\widehat{p}_{\text{gpc}}$ the with NLML discrepancy. We see that the FPR is stable across ICL dataset size n and that the FPR aligns well with the significance level α. Figure 15c plots the FPR for the capability detector defined by $\widehat{p}_{\text{gpc}}$ the with NLL discrepancy. In contrast, the FPR decreases with increasing ICL dataset size.

Figure 16a shows p-values under the Llama-2 7B model as a function of the ICL dataset x^n size n (context length). Again, we see a clear separation between the estimated generative predictive p-values $\widehat{p}_{\text{gpc}}$ for the in-capability (solid lines) and the out-of-capability (dashed lines) tasks. The separation is robust across different ICL dataset sizes and the two discrepancy functions we test.

Figure 16b plots the FPR for the capability detector defined by $\widehat{p}_{\text{gpc}}$ the with NLML discrepancy. We see that the FPR trends up with increasing ICL dataset size n and approaches the significance level α. Figure 16c plots the FPR for the capability detector defined by $\widehat{p}_{\text{gpc}}$ the with NLL discrepancy. As for



(a) Generative predictive p-values (b) False Positive Rate (NLML) (c) False Positive Rate (NLL)

Figure 16: Natural language task. The generative predictive p-value against dataset size n and it's relationship to the false positive rate. The first figure shows the generative predictive p-values, and the second and third figures show the false positive rate with the NLML and NLL discrepancy functions, respectively.

the simulated regression data, the FPR starts to decrease with increasing ICL dataset size, but for lower α values the trend begins to reverse.

G Choice of discrepancy function

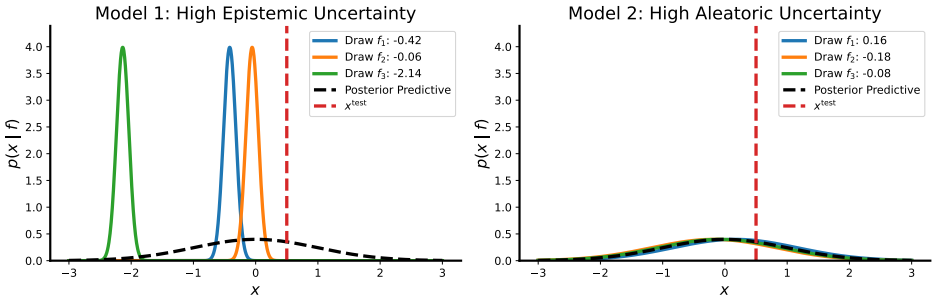


Figure 17: Comparison of two models: Model 1 exhibits high epistemic uncertainty, while Model 2 exhibits high aleatoric uncertainty. The posterior predictive distribution is shown alongside samples of f and the test point x^{test}. Note that the numerical values in the plot differ from the example in the text for visual clarity. The discrepancy function $g(x, f)$ helps distinguish these cases by evaluating how likely x^{test} is under different values of f.

It might not be immediately clear when to use one discrepancy function over another. In this section we argue that in general one should prefer to use $g(\mathrm{x}, \mathrm{f})$ when possible as it can identify when the model has high uncertainty about the value of f.

To illustrate this difference, imagine two models. The first model has posterior $p_1(f \mid x^n) = N(0, 1)$ and likelihood $p_1(x \mid f) = N(f, 0.0001)$. The second model has posterior $p_2(f \mid x^n) = N(0, 0.0001)$ and likelihood $p_2(x \mid f) = N(f, 1)$. In both models, the posterior predictive is essentially the same standard normal, $p(x \mid x^n) = N(0, 1)$. However, in the first model, the posterior predictive variance is due to epistemic uncertainty (we are unsure about the correct f), while in the second model, it is due to aleatoric uncertainty (we are confident about f, but the task is inherently stochastic). Figure Figure 17 illustrates this scenario.

Now assume that we have a test point $x^{\text{test}} = 0.5$. Ideally, we want to say that the second model does a good job of predicting this data point because the task is well specified, while the first one does not because we are still uncertain about the task. That is, the first model assigns high probability to many values of f where x^{test} is unlikely.

Depending on the discrepancy we choose, we may or may not be able to distinguish between the two scenarios and reject the correct model. For example, if we use

$$g(x, x^n) := -\sum_{z_i, y_i \in x} \log p(z_i, y_i \mid x^n),$$

$\log p(z_i, y_i \mid x^n)$ is the same for both models and we will have identical p-values, which will indicate that both models are suitable—an undesirable outcome.

On the other hand, if we use

$$g(x, f) := -\sum_{z_i, y_i \in x} \log p(z_i, y_i \mid f),$$

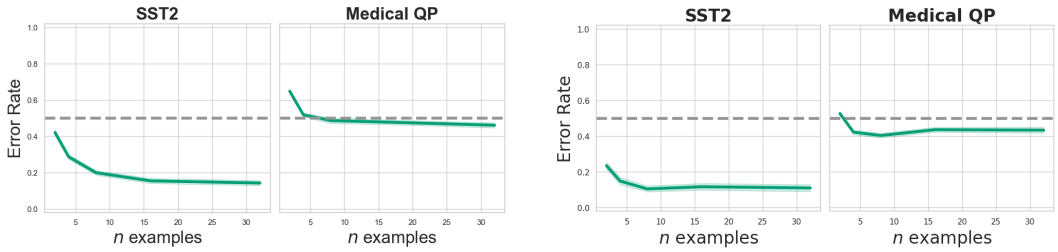
the p-values will be quite different. For the first model, many values of f will fall far away from x^{test} when computing

$$p_{\text{ppc}} := \int\int \mathbb{1}\{g(x, f) \geq g(x^{\text{test}}, f)\}\, dP(x \mid f)\, dP(f \mid x^n).$$

As a result, $g(x^{\text{test}}, f)$ will be much lower than $g(x, f)$, and the PPC will be quite low. However, in the second model, this will not happen, as all values of f will be sampled around 0, and x^{test} is a reasonable observation for a normal distribution centered at $f \approx 0$ with standard deviation 1. Therefore, using the f-dependent PPC provides the desired behavior because it informs us when the model is confused about the task. Although we have used an example for illustrative purposes, the same behavior holds generally.

H GEMMA-2 9B RESULTS

Figure 18b shows that the in-capability vs. out-of-capability distinction is also sensible for the Gemma-2 9B model. So we conduct the same analysis for Gemma-2 9B that we did for Llama-2 7B in Section 7.



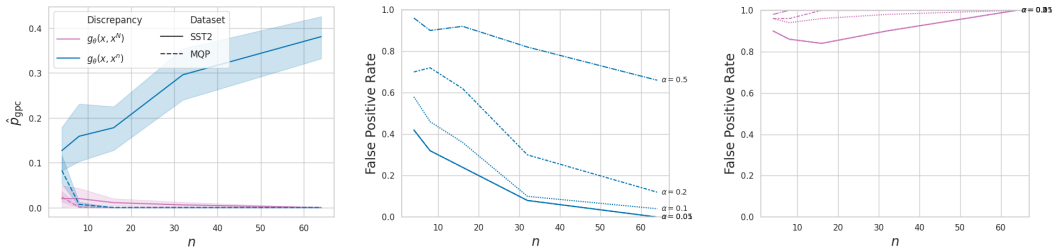
(a) Llama 2 7B

(b) Gemma 2 9B

Figure 18: Natural language in-capability vs. out-of-capability tasks.

Figure 19a shows p-values under the Gemma-2 9B model as a function of the ICL dataset x^n size n (context length). We see a clear separation between the estimated generative predictive p-values $\widehat{p}_{\text{gpc}}$ for the in-capacity SST2 data (solid lines) and the out-of-capacity MQP dataset (dashed lines), but only for the NLML discrepency. The separation is robust across different ICL dataset sizes.

Figure 19b plots the FPR for the capability detector defined by $\widehat{p}_{\text{gpc}}$ the with NLML discrepancy. We do not see the same stability of the FPR across ICL dataset size n that we saw for the Llama-2 7B model. Instead the FPR decreases with increasing n for all significance level α. Figure 19c plots the FPR for the capability detector defined by $\widehat{p}_{\text{gpc}}$ the with NLL discrepancy. We see that the false positive rate is high for all values. These findings are reflected in the Precision curves on the left hand side of Figure 20. We again see in the Recall curve that the NLL discrepancy leads to a more sensitive predictor than the NLML discrepancy. The F1 and Accuracy curves show that the NLML based p-value leads to a much more effective predictor for Gemma-2 9B.



(a) Generative predictive p-values (b) False Positive Rate (NLML) (c) False Positive Rate (NLL)

Figure 19: Natural language task with Gemma-2 9B. The generative predictive p-value against dataset size n and it's relationship to the false positive rate. The first figure shows the generative predictive p-values, and the second and third figures show the false positive rate with the NLML and NLL discrepancy functions, respectively.

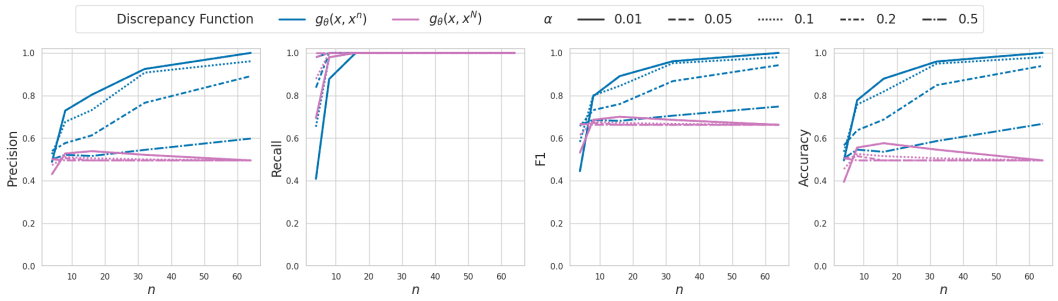


Figure 20: Natural language model suitability detection ablation. Precision, recall, F1, and accuracy metrics vs. number of in-context examples. SST2 ICL datasets are taken to be in-capability for Gemma-2 9b. MQP ICL datasets are taken to be out-of-capability for Gemma-2 9b.