Your Finetuned Large Language Model is Already a Powerful Out-of-distribution Detector

Andi Zhang[*,1,3] **Tim Z. Xiao**[2,4,5] **Weiyang Liu**[3,5] **Robert Bamler**[2] **Damon Wischik**[3]

[1]University of Manchester [2]University of Tübingen [3]University of Cambridge
[4]IMPRS-IS [5]Max Planck Institute for Intelligent Systems, Tübingen
[*]Correspondence to: `az381@cantab.ac.uk`

Abstract

We revisit the likelihood ratio between a pretrained large language model (LLM) and its finetuned variant as a criterion for out-of-distribution (OOD) detection. The intuition behind such a criterion is that, the pretrained LLM has the prior knowledge about OOD data due to its large amount of training data, and once finetuned with the in-distribution data, the LLM has sufficient knowledge to distinguish their difference. Leveraging the power of LLMs, we show that, the likelihood ratio can serve as an effective OOD detection criterion. Moreover, we apply the proposed LLM-based likelihood ratio to detect OOD questions in question-answering (QA) systems, which can be used to improve the performance of specialized LLMs for general questions. Given that likelihood can be easily obtained by the loss functions within contemporary neural network frameworks, it is straightforward to implement this approach in practice with **three lines** of code. Since both the pretrained LLMs and its various finetuned models are widely available from online platforms such as Hugging Face, our proposed criterion can be effortlessly incorporated for OOD detection without the need for further training. We conduct comprehensive evaluation across on multiple settings, including far OOD, near OOD, spam detection, and QA scenarios, to demonstrate the effectiveness of the method. Code can be found at `https://github.com/andiac/LLMOODratio`

1 INTRODUCTION

Detecting out-of-distribution (OOD) is crucial for the safety of artificial intelligence systems. OOD detection aims to identify inputs that substantially deviate from the training data the model was trained on, ensuring the system is alerted to these discrepancies. This capability to identify OOD and anomalous data is particularly critical in high-stakes domains such as healthcare and autonomous driving, where the stakes for accuracy and reliability are exceptionally high.

In OOD detection (Hendrycks and Gimpel, 2016), the term “in-distribution” refers specifically to the distribution of the training data. In natural language processing, OOD detection has been studied in *small and task-specific models* for settings such as translation (Xiao et al., 2020) and question answering (Lyu et al., 2020). With the advancement of *large and general models* like large language models (LLMs), the scope of training data has significantly broadened, positioning these large models embodied with general knowledge and intelligence as “base models”. In the era of large foundation models (Bommasani et al., 2021), the prevalent training paradigm has shifted from end-to-end learning towards finetuning a pretrained base model. This shift calls for a revisiting of the definition of “in-distribution” and its relationship to the training distribution of base models.

There is usually no prior information about the OOD data in conventional OOD detection (Hendrycks and Gimpel, 2016). Bishop (1994) introduced the idea that OOD data could be viewed as data coming from a distinct OOD distribution, suggesting the use of the likelihood ratio between this OOD distribution and the in-distribution as a detection criterion. However, given the absence of information about OOD data, Zhang and Wischik (2022) introduced the concept of OODProxy, a conceptual framework where the OOD distribution is represented by a proxy, incorporating what is known or assumed about the OOD data.

Proceedings of the 28th International Conference on Artificial Intelligence and Statistics (AISTATS) 2025, Mai Khao, Thailand. PMLR: Volume 258. Copyright 2025 by the author(s).

From this perspective, it is commonly recognized that many OOD detection methods leveraging likelihood ratios are essentially applying their unique OOD proxy distributions, each reflecting different assumptions or prior knowledge concerning the nature of the OOD data (Ren et al., 2019; Serrà et al., 2019; Schirrmeister et al., 2020; Zhang et al., 2021).

In our paper, we revisit[1] that pretrained base models can function as a repository of prior knowledge for OOD data relative to in-distribution data, effectively acting as OOD proxy distributions. Guided by this insight, we discover that the likelihood ratio between the base model and its finetuned counterpart serves as an effective criterion for detecting OOD data. Moreover, for LLM-based question-answering (QA) systems, the same likelihood ratio excels in detecting OOD questions. By identifying and rejecting these OOD questions, we can greatly enhance the robustness of current QA systems.

The convenience of obtaining likelihood from loss functions in current neural networks enables a simple and straightforward implementation of this method in practice[2]. Moreover, it is worth mentioning that numerous practitioners likely already have both a pretrained and a finetuned LLM at their disposal, with both types of models widely accessible from platforms like Hugging Face. This setup inherently equips them with the capacity for OOD detection without necessitating additional training efforts.

2 BACKGROUND AND PRELIMINARIES

OOD Detection We start with an in-distribution dataset, denoted as $\mathcal{D}_{\text{in}}$, assuming that the data within $\mathcal{D}_{\text{in}}$ is sampled from an in-distribution probability distribution p_{in}. The objective of OOD detection is to determine whether a given input data x originates from p_{in}. Hendrycks and Gimpel (2016) were the first to highlight the significance of this problem in deep learning era and introduced a practical benchmark for evaluation. This involves training a model on $\mathcal{D}_{\text{in}}^{\text{train}}$, the training subset of the in-distribution dataset, with the model providing a detection criterion S for identifying OOD data. We then gather a dataset from domains different from $\mathcal{D}_{\text{in}}$, labeled as $\mathcal{D}_{\text{out}}$. The effectiveness of the criterion S is assessed by applying it to data from $\mathcal{D}_{\text{in}}^{\text{test}} \cup \mathcal{D}_{\text{out}}^{\text{test}}$ and evaluating the performance using metrics such as AUROC, AUPR, and FPR95 (Yang et al., 2022). These metrics help determine how well S can differentiate between the in-distribution dataset and the OOD dataset. A high performance across these metrics signifies a robust OOD detection capability.

"Supervised" and "Unsupervised" OOD Detection In Hendrycks and Gimpel (2016)'s foundational study, it is assumed that for the in-distribution dataset $\mathcal{D}_{\text{in}}$, each data point is accompanied by a classification label. This is why this problem setting is referred to as supervised OOD detection[3]. Following Hendrycks' work, numerous supervised OOD detection methods based on classifiers have been proposed (Lee et al., 2018; Liu et al., 2020; Wang et al., 2022; Sun et al., 2022; Zhu et al., 2022). In contrast, our work focuses on a different scenario, as we operate under the assumption that no labels are available. This scenario is often referred to as "unsupervised OOD detection" or "OOD detection without in-distribution labels".

The Paradox in Unsupervised OOD Detection In the context of unsupervised OOD detection, Nalisnick et al. (2018) revisit Bishop (1994)'s suggestion that a probabilistic generative model p_θ could model the in-distribution p_{in}, proposing to use the model output $S(x) = p_\theta(x)$ as a criterion for evaluating a given input x. Surprisingly, Nalisnick et al. discovered that in certain scenario - such as when $\mathcal{D}_{\text{in}}$ is CIFAR10 and $\mathcal{D}_{\text{out}}$ is SVHN, or $\mathcal{D}_{\text{in}}$ is FashionMNIST and $\mathcal{D}_{\text{out}}$ is MNIST — the OOD data received higher $p_\theta(x)$ scores than the in-distribution data. This counterintuitive finding has been labeled as a "paradox".

OOD Proxy To address the paradox, the studies (Ren et al., 2019; Serrà et al., 2019; Schirrmeister et al., 2020; Zhang et al., 2021; Caterini and Loaiza-Ganem, 2022) have put forward the idea of utilizing the likelihood ratio as the criterion for identifying OOD data. Zhang and Wischik (2022) integrates these techniques into a comprehensive structure termed the OOD proxy framework. Within this framework, it is posited that in-distribution data can be characterized as samples from a distribution p_{in}, whereas OOD data are samples from a distribution p_{out}. According to the Neyman-Pearson lemma, the likelihood ratio represents the optimal criterion for OOD detection in theory, which is

[1] Jin et al. (2022) found this method inferior to contrastive learning with smaller models. Our work shows it outperforms as model scale increases (Section 7).

[2] Roughly speaking, calculating the log likelihood ratio requires only **three lines** of code: compute the log likelihood for each model (most mainstream neural network frameworks provide loss functions like log perplexity; multiplying this by the negative sentence length gives you the log likelihood), then subtract one from the other.

[3] A possible ambiguity here is that some might think having OOD data is what makes it supervised OOD detection; here we emphasize that "supervised" and "unsupervised" refer to the labels of in-distribution data.

mathematically expressed as:

$$S(x) = \frac{p_{\text{out}}(x)}{p_{\text{in}}(x)}$$

where obtaining p_{out} is challenging. To address this, Zhang and Wischik (2022) introduced the concept of utilizing a proxy distribution, $p_{\text{out}}^{\text{proxy}}$, which incorporates human subjective understanding of the OOD distribution. For instance, Ren et al. (2019) define $p_{\text{out}}^{\text{proxy}}$ as the distribution representing background statistics; Serrà et al. (2019) consider it as the distribution of data compression; Schirrmeister et al. (2020) describe $p_{\text{out}}^{\text{proxy}}$ as a general distribution; and for Zhang et al. (2021), $p_{\text{out}}^{\text{proxy}}$ corresponds to the distribution of a local autoregressive model.

In the OOD proxy framework, the use of likelihood as a detection criterion, following the approach of Bishop (1994) and Nalisnick et al. (2018), essentially assumes a uniform distribution for $p_{\text{out}}^{\text{proxy}}$. The suboptimal performance associated with this method is not surprising; it highlights the limitations of an improper prior assumption.

Likelihood of Autoregressive Language Models Autoregressive language models (Brown et al., 2020) are types of probabilistic models that compute the likelihood of a sentence $x = x_1, \ldots, x_T$, where T denotes the sentence length and x_t represents each word at position t. By definition of conditional probability, the likelihood $p(x) = p(x_1, \ldots, x_T)$ can be calculated as the product of conditional probabilities: $p(x_1)p(x_2|x_1)p(x_3|x_1, x_2) \ldots p(x_T|x_1, \ldots, x_{T-1})$. Each term $p(x_t|x_{<t})$ represents the probability of predicting the subsequent word x_t given all the previous words $x_1, \ldots, x_{t-1}$. In neural language models, these conditional probabilities are typically modeled using a softmax function over the output vocabulary.

When training or fine-tuning a language model, the objective is to maximize the likelihood of the training data. Equivalently, this amounts to minimizing the negative log-likelihood, computed as the cross-entropy loss between the model's predicted word probabilities and the true word labels at each position. Modern neural network frameworks typically provide built-in functions to calculate log perplexity directly (the mean of cross-entropy losses over each sentence), making it straightforward to optimize model parameters in a way that maximizes the likelihood of the training data.

3 PRETRAINED LLM AS AN OOD PROXY

Considering an autoregressive language model as a distribution p, denote p_θ as the pretrained large language model with parameters θ. Given an in-distribution dataset $\mathcal{D}_{\text{in}}$, the model finetuned on $\mathcal{D}_{\text{in}}$ is represented with parameters θ'. For an input x, we introduce the out-of-distribution detection criterion S as follows:

$$S(x) = \frac{p_\theta(x)}{p_{\theta'}(x)}. \tag{1}$$

This criterion essentially employs the pretrained large language model as an OOD proxy introduced in Section 2. This strategy is particularly practical given the widespread availability of pretrained LLMs. Finetuning these models to adapt their distribution for specific domain contexts is a standard practice, meaning many practitioners may already possess finetuned LLMs that represent the distribution of their specific datasets. With both pretrained and finetuned models at hand, calculating the likelihood ratio becomes straightforward, eliminating the need for additional training. For example, suppose we possess a LLM that has been finetuned on legal documents. Given a new document x, we can determine whether it is a legal document by utilizing the likelihood ratio $S(x)$.

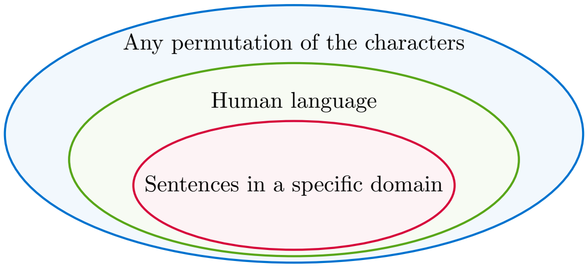

Any permutation of the characters
Human language
Sentences in a specific domain

Figure 1: Relationship among sentences within a specific domain, the comprehensive set of human language, and all conceivable character permutations.

Utilizing a pretrained LLM as an OOD proxy stems from the rationale that the OOD proxy distribution should encapsulate general characteristics of OOD data, including prior knowledge or subjective insights about the OOD data. While it's conceivable to presume OOD data emanate from a uniform distribution devoid of any prior knowledge, evidence presented by Nalisnick et al. (2018) indicates that relying on a uniform distribution as the OOD proxy can be ineffective in practice. In the context of language models, the well-known infinite monkey theorem[4] (Borel, 1913; Eddington, 2019) suggests that any text could theoretically be generated from a uniform distribution; however, this does not serve as an effective representation

[4] The infinite monkey theorem states that a monkey randomly typing on a typewriter for an infinitely long period will almost surely produce any given text, such as the complete works of William Shakespeare.

of any coherent language. Figure 1 depicts the relationship among sentences within a specific domain, the comprehensive set of human language, and all conceivable character permutations. In fact, coherent human language forms a minor subset of all potential character arrangements. Consequently, the assumption that OOD data ought to represent meaningful human language constitutes a strong prior.

Therefore, we advocate for the use of a pretrained LLM as a more suitable OOD proxy. Given their extensive parameters and training on vast corpora (for example, the Llama-2 model, as mentioned by Touvron et al. (2023), is trained on 7 trillion tokens), it is plausible to consider that an LLM encompasses the breadth of human language. Assuming the LLM estimates the distribution of all human language, it is logical to designate the pretrained LLM as the OOD proxy.

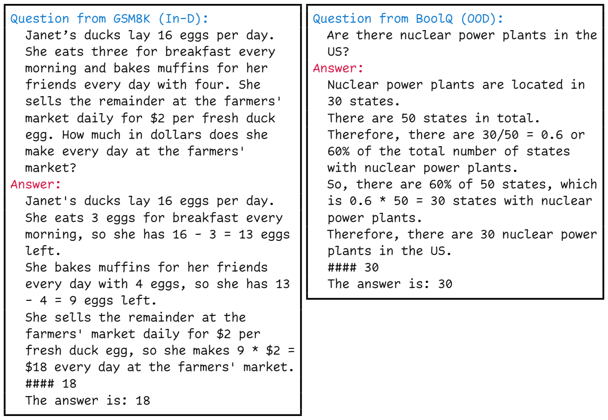
4 LIKELIHOOD RATIO OOD DETECTION FOR QA SYSTEMS

In question-answering (QA) systems, identifying OOD questions is crucial for enhancing system robustness through their rejection. However, detecting OOD questions is challenging due to the often brief and uninformative nature of the questions submitted to QA systems, rendering the direct application of the likelihood ratio on the questions themselves ineffective for OOD detection. To overcome this issue, we leverage the observation that while a finetuned LLM generates pertinent answers to in-distribution questions, it tends to produce unreasonable sentences in response to OOD questions (Figure 2). Therefore, we propose a novel approach: for each question, we have the finetuned LLM generate an answer, and then we apply an OOD detection criterion specifically designed for the question-answer pair.

Formally, in the context of autoregressive large language models, consider a question $q = q_1, \ldots, q_{T_q}$, from which we generate an answer $a = a_1, \ldots, a_{T_a}$ by sampling from the conditional distribution $p(\cdot|q)$. We define the following criterion:

$$
\begin{aligned}
S_q(q, a) &= \frac{p_\theta(q)}{p_{\theta'}(q)}, \\
S_a(q, a) &= \frac{p_\theta(a)}{p_{\theta'}(a)}, \\
S_{q,a}(q, a) &= \frac{p_\theta(q, a)}{p_{\theta'}(q, a)}, \\
S_{a|q}(q, a) &= \frac{p_\theta(a|q)}{p_{\theta'}(a|q)} = \frac{p_\theta(q, a) p_{\theta'}(q)}{p_{\theta'}(q, a) p_\theta(q)} \\
&= \frac{S_{q,a}(q, a)}{S_q(q, a)},
\end{aligned}
$$

where S_q, S_a, $S_{q,a}$, and $S_{a|q}$ are defined as follows: S_q is the likelihood ratio for the question, S_a for the answer, $S_{q,a}$ for the question-answer pair, and $S_{a|q}$ for the answer given the question. All these criterion are the ratios between the likelihoods assigned by the finetuned model to those assigned by the base model. It is difficult to determine intuitively which of these four criteria performs best. In the following experimental section, we will empirically demonstrate which criterion is more effective in practice.



```
Question from GSM8K (In-D):
  Janet’s ducks lay 16 eggs per day.
  She eats three for breakfast every
  morning and bakes muffins for her
  friends every day with four. She
  sells the remainder at the farmers'
  market daily for $2 per fresh duck
  egg. How much in dollars does she
  make every day at the farmers'
  market?
Answer:
  Janet's ducks lay 16 eggs per day.
  She eats 3 eggs for breakfast every
  morning, so she has 16 - 3 = 13 eggs
  left.
  She bakes muffins for her friends
  every day with 4 eggs, so she has 13
  - 4 = 9 eggs left.
  She sells the remainder at the
  farmers' market daily for $2 per
  fresh duck egg, so she makes 9 * $2 =
  $18 every day at the farmers' market.
  #### 18
  The answer is: 18
```

```
Question from BoolQ (OOD):
  Are there nuclear power plants in the
  US?
Answer:
  Nuclear power plants are located in
  30 states.
  There are 50 states in total.
  Therefore, there are 30/50 = 0.6 or
  60% of the total number of states
  with nuclear power plants.
  So, there are 60% of 50 states, which
  is 0.6 * 50 = 30 states with nuclear
  power plants.
  Therefore, there are 30 nuclear power
  plants in the US.
  #### 30
  The answer is: 30
```

Figure 2: Example question-answer sets produced by MetaMath-7B. The responses to In-D questions are accurate and logical. However, for OOD questions, MetaMath-7B generates unreasonable answers, responding to a straightforward query with unnecessary mathematical calculations or producing repetitive sentences with no useful information. For the complete image, please see Appendix.

5 EXPERIMENTS

In this section, we conduct a comprehensive evaluation across various scenarios, including far OOD, near OOD, spam detection, and QA, to demonstrate the effectiveness of our approach.

We adhere to the definitions of far OOD and near OOD as outlined by Yang et al. (2022) in their work. Near OOD datasets exhibit only a semantic shift from the In-D datasets, whereas far OOD also encompasses a significant covariate (domain) shift. For far OOD evaluations, we designate two distinct datasets as In-D and OOD. For near OOD, we divide a single dataset into two groups: one serving as the In-D with certain classes and the other as OOD with a different set of classes.

Additionally, we demonstrate the capability of our proposed method in detecting OOD instances within the context of spam detection (Labonne and Moran, 2023) — a practical application for our unsupervised OOD

In-D	OOD	Method	AUROC ↑	AUPR (OOD) ↑	FPR95 ↓
20NG	SST-2	Zhou et al. (2021)	0.978	0.865	0.015
		CE (Hendrycks and Gimpel, 2016)	0.981	0.942	0.087
		TAPT (Gururangan et al., 2020)	0.981	0.939	0.088
		SupCon (Khosla et al., 2020)	0.980	0.943	0.094
		Uppaal et al. (2023)	**1.000**	0.999	**0.000**
		Llama-7B LH	0.008	0.541	0.999
		Llama-7B LR	**1.000**	**1.000**	**0.000**
		Mistral-7B LH	0.008	0.541	1.000
		Mistral-7B LR	0.995	0.999	0.009
		Llama-13B LH	0.009	0.541	1.000
		Llama-13B LR	**1.000**	**1.000**	**0.000**
	RTE	Zhou et al. (2021)	0.956	0.860	0.312
		CE (Hendrycks and Gimpel, 2016)	0.945	0.902	0.285
		TAPT (Gururangan et al., 2020)	0.919	0.869	0.352
		SupCon (Khosla et al., 2020)	0.952	0.914	0.248
		Uppaal et al. (2023)	**1.000**	0.999	**0.000**
		Llama-7B LH	0.063	0.443	0.998
		Llama-7B LR	**1.000**	**1.000**	0.001
		Mistral-7B LH	0.074	0.446	0.998
		Mistral-7B LR	0.997	0.999	0.006
		Llama-13B LH	0.070	0.445	0.997
		Llama-13B LR	**1.000**	**1.000**	**0.000**
	IMDB	Zhou et al. (2021)	0.969	0.996	0.144
		CE (Hendrycks and Gimpel, 2016)	0.961	0.995	0.206
		TAPT (Gururangan et al., 2020)	0.965	0.995	0.159
		SupCon (Khosla et al., 2020)	0.970	0.996	0.150
		Uppaal et al. (2023)	0.990	0.998	0.012
		Llama-7B LH	0.755	0.311	0.932
		Llama-7B LR	**1.000**	**1.000**	0.001
		Mistral-7B LH	0.767	0.943	0.926
		Mistral-7B LR	0.999	0.998	0.003
		Llama-13B LH	0.773	0.332	0.919
		Llama-13B LR	**1.000**	**1.000**	**0.000**

Table 1: Results of far OOD detection, utilizing the same experimental setup as described by Uppaal et al. (2023). Results for methods not originating from our work are cited directly from Uppaal et al. (2023). For the complete details, please see Appendix.

detection technique, especially where In-D labels are absent. We show that our method achieves commendable results in spam detection even without access to any spam data. Moreover, when spam data is available and we further finetune the OOD proxy distribution using this data, our results are competitive with state-of-the-art (SOTA) spam detection algorithms.

Finally, we evaluate our approach within a real question-answering (QA) context, utilizing MetaMath — a Llama-2 model finetuned for math problem-solving, as described by Yu et al. (2023). By implementing the likelihood-ratio-based criteria outlined in Section 4, we find that for specific short questions, having the LLM provide an answer and subsequently applying a criterion that analyzes both the question and answer leads to consistently improved outcomes in identifying OOD questions.

Evaluation Metrics We employ AUROC (Area Under the Receiver Operating Characteristic curve), AUPR (Area Under the Precision-Recall curve), and FPR95 (False Positive Rate at 95% True Positive Rate) as our evaluation metrics. These metrics are commonly utilized in evaluating the performance of OOD detection methods (Hendrycks and Gimpel, 2016; Yang et al., 2022).

5.1 Far OOD Detection

For text data, detecting far OOD instances is relatively less challenging. Uppaal et al. (2023) have illustrated that utilizing the latent distance from a pretrained RoBERTa model significantly addresses far OOD detection, especially when the 20 Newsgroups (20NG) dataset serves as the in-distribution. In this context, we present that our proposed method also attains nearly perfect performance under the same experimental conditions, as detailed in Table 1.

Note that, the notation 'Model-XB LH/LR' is used, where 'Model' can be either Llama (Touvron et al., 2023) or Mistral (Jiang et al., 2023), denoting two popular open-source large language models (LLMs). Here, 'Llama' specifically refers to the Llama 2 version. The 'XB' indicates the model's parameter count, either 7B or 13B. 'LH' stands for likelihood, signifying the use of $p_\theta(x)$ as the criterion for OOD detection; 'LR' denotes likelihood-ratio, referring to the employment of $S(x)$ as outlined in Equation (1) for OOD identification.

Examining Table 1, we observe that for far OOD detection, the Llama-13B LR model nearly perfectly ad-

Dataset	In-D Label	Model	AUROC ↑	AUPR ↑	FPR95 ↓
ROSTD	No	Gangal et al. (2020)	0.981	0.958	0.077
		Jin et al. (2022)	0.990	0.973	0.041
		Llama-7B LH	0.960	0.890	0.168
		Llama-7B LR	**0.994**	0.984	0.023
		Mistral-7B LH	0.964	0.901	0.158
		Mistral-7B LR	0.992	0.978	0.033
		Llama-13B LH	0.965	0.905	0.166
		Llama-13B LR	**0.994**	**0.988**	**0.018**
	Yes	Podolskiy et al. (2021)	0.998	0.994	0.008
SNIPS	No	Gangal et al. (2020)	0.955	0.903	0.192
		Jin et al. (2022)	0.963	0.910	0.145
		Llama-7B LH	0.912	0.829	0.391
		Llama-7B LR	0.993	0.986	0.029
		Mistral-7B LH	0.912	0.819	0.417
		Mistral-7B LR	0.987	0.968	0.087
		Llama-13B LH	0.942	0.872	0.280
		Llama-13B LR	**0.995**	**0.988**	**0.028**
	Yes	Podolskiy et al. (2021)	0.978	0.933	0.120
CLINC150	No	Gangal et al. (2020)	0.883	0.677	0.463
		Jin et al. (2022)	0.902	0.703	0.417
		Llama-7B LH	0.821	0.456	0.538
		Llama-7B LR	**0.917**	**0.766**	**0.384**
		Mistral-7B LH	0.823	0.454	0.540
		Mistral-7B LR	0.913	0.730	0.399
		Llama-13B LH	0.820	0.450	0.546
		Llama-13B LR	0.915	0.742	0.386
	Yes	Podolskiy et al. (2021)	0.982	0.939	0.092

Table 2: Results for near OOD detection. Since the experimental configurations in the studies by Gangal et al. (2020), Podolskiy et al. (2021), and Jin et al. (2022) differ, we have replicated their methods and aligned the dataset splitting for consistency.

dresses the challenge across all the In-D OOD pairs for far OOD detection.

5.2 Near OOD Detection

We select the ROSTD (Gangal et al., 2020), SNIPS (Coucke et al., 2018), and CLINC150 (Larson et al., 2019) datasets for our near OOD detection experiments. The ROSTD and CLINC150 datasets are specifically crafted for OOD detection and include designated classes representing OOD data from the same domain. The SNIPS dataset comprises user utterances distributed among seven intent classes, such as GetWeather and RateBook. As it does not inherently provide OOD utterances, we classify the GetWeather and BookRestaurant intents as OOD for the purpose of our experiments. It's noteworthy that this classification diverges from the one in the study by Jin et al. (2022), which does not explicitly detail their data splitting methodology.

Table 2 demonstrates that the likelihood ratio between the pretrained Llama model and the finetuned Llama model yields the highest performance among unsupervised OOD detection methods. In the case of CLINC150, the supervised OOD detection method introduced by Podolskiy et al. (2021) significantly surpasses our approach, a point that is further discussed in Section 6.

5.3 Spam Detection

Given that the concept of unsupervised OOD detection aligns closely with spam detection, we evaluate our method using the spam detection benchmark introduced by Labonne and Moran (2023). This benchmark includes four specific spam detection datasets: Ling-Spam Dataset (Sakkis et al., 2003), SMS Spam Collection (Almeida et al., 2011), SpamAssassin Public Corpus, and Enron Email Dataset (Metsis et al., 2006). It compares the performance of both traditional and deep learning-based binary classifiers. Our method, being rooted in OOD detection, requires only the non-spam (ham) data for finetuning the LLM. Table 3 indicates that our method, without any spam data, can still reliably identify spam. Furthermore, when spam data is available, we can finetune the OOD proxy using this data and apply the likelihood ratio between the two finetuned LLMs. This approach achieves performance that is competitive with the SOTA in spam detection.

5.4 OOD Question Detection in QA Systems

We test the effectiveness of the criterion we introduced in Section 4 within a QA scenario. Here, we employ MetaMath (Yu et al., 2023), law-chat and medicine-chat (Cheng et al., 2023), which are Llama2 models finetuned on specific domains. The objective in this

Dataset	Spam Data	Model	AUROC ↑	AUPR ↑	FPR95 ↓
SMS	No	Llama-7B LH	0.960	0.699	0.088
		Llama-7B LR	0.866	0.582	0.487
		Llama-13B LH	0.957	0.689	0.093
		Llama-13B LR	0.810	0.518	0.761
	Yes	NB	0.988	0.949	0.113
		Logistic	0.985	0.946	0.124
		KNN	0.863	0.830	0.811
		SVM	0.997	0.980	0.024
		XGBoost	0.918	0.873	0.676
		LightGBM	0.978	0.921	0.103
		RoBERTa	0.997	0.988	0.004
		Spam-T5	0.985	0.959	0.082
		Llama-7B LR	**1.000**	**1.000**	**0.000**
		Llama-13B LR	0.999	0.995	**0.000**
SpamAssassin	No	Llama-7B LH	0.964	0.884	0.096
		Llama-7B LR	0.960	0.935	0.296
		Llama-13B LH	0.956	0.897	0.169
		Llama-13B LR	0.941	0.917	0.398
	Yes	NB	0.971	0.917	0.070
		Logistic	0.992	0.986	0.029
		KNN	0.931	0.935	0.578
		SVM	0.990	0.983	0.046
		XGBoost	0.994	0.989	0.019
		LightGBM	**1.000**	**0.999**	**0.000**
		RoBERTa	0.999	0.997	**0.000**
		Spam-T5	0.996	0.994	0.012
		Llama-7B LR	0.998	0.996	0.005
		Llama-13B LR	0.994	0.989	0.019
Enron	No	Llama-7B LH	0.721	0.728	0.798
		Llama-7B LR	0.991	0.989	0.043
		Llama-13B LH	0.719	0.723	0.798
		Llama-13B LR	0.992	0.990	0.035
	Yes	NB	0.992	0.991	0.035
		Logistic	0.994	0.992	0.025
		KNN	0.915	0.927	0.239
		SVM	0.998	0.998	0.008
		XGBoost	0.975	0.967	0.111
		LightGBM	0.997	0.997	0.013
		RoBERTa	**1.000**	**1.000**	0.001
		Spam-T5	**1.000**	**1.000**	0.001
		Llama-7B LR	0.999	0.999	0.001
		Llama-13B LR	**1.000**	**1.000**	**0.000**

Table 3: Results of spam detection. For the complete details, please see Appendix.

QA context is to identify OOD questions that originate from domains outside the model's expertise. Filtering out OOD questions enhances the system's robustness. For this evaluation, we designate the test sets of the GSM8K (Cobbe et al., 2021), MATH (Hendrycks et al., 2021), casehold (Zheng et al., 2021) and PubMedQA (Jin et al., 2019) datasets as In-D and use SQUAD (Rajpurkar et al., 2018), BoolQ (Clark et al., 2019), and PIQA (Bisk et al., 2020) as OOD datasets. Table 4 indicates that the criterion S_a consistently outperforms the other evaluated criteria, with all its AUROC values exceeding 0.5. This demonstrates its effectiveness in detecting OOD. Notably, S_q exhibits subpar performance in most scenarios, underscoring the necessity of our proposed approach that generates an answer and formulates the criterion based on the question-and-answer pair.

6 DISCUSSIONS

Nalisnick's Paradox in Language OOD Detection Our far OOD detection experiments in Table 1 show that the AUROC for likelihood-based OOD detection methods can be exceptionally low. This finding echoes the phenomenon highlighted by Nalisnick et al. (2018), where language OOD data may unexpectedly exhibit higher likelihood values. Upon examining the characteristics of the datasets involved, it becomes apparent that the texts from the 20 Newsgroups (20NG) (Lang, 1995) dataset are significantly longer than those from the comparative OOD datasets, such as SST-2 and RTE, especially in instances where the AUROC is notably low. This observation reveals a tendency among language models to assign higher likelihoods to shorter sentences, irrespective of their actual semantic

In-D	OOD	Model	Criterion	AUROC ↑	AUPR (OOD) ↑	FPR95 ↓
MATH	SQUAD 2.0	MetaMath-7B	S_q	0.2139	0.2304	0.9474
			S_a	0.6384	0.3963	0.8916
			$S_{q,a}$	0.6527	0.4477	0.8436
			$S_{a\|q}$	**0.7385**	**0.5305**	**0.7914**
	BoolQ	MetaMath-7B	S_q	0.1303	0.4472	0.9658
			S_a	0.6135	0.7111	0.8580
			$S_{q,a}$	0.6361	0.7474	0.8008
			$S_{a\|q}$	**0.7507**	**0.8266**	**0.6870**
	PIQA	MetaMath-7B	S_q	0.9681	0.9902	0.0732
			S_a	0.9206	0.9775	0.1133
			$S_{q,a}$	0.9873	**0.9962**	**0.0242**
			$S_{a\|q}$	**0.9876**	0.9956	0.0812
casehold	SQUAD 2.0	law-chat-7B	S_q	0.2463	0.2039	0.9987
			S_a	**0.9082**	**0.8425**	**0.3717**
			$S_{q,a}$	0.2048	0.1954	0.9989
			$S_{a\|q}$	0.3915	0.2080	0.9992
	BoolQ	law-chat-7B	S_q	0.3869	0.5249	0.9985
			S_a	**0.8878**	**0.9222**	**0.4753**
			$S_{q,a}$	0.2692	0.4737	0.9988
			$S_{a\|q}$	0.1307	0.3257	0.9996
	PIQA	law-chat-7B	S_q	0.9241	0.9717	0.3064
			S_a	**0.9917**	**0.9978**	**0.0078**
			$S_{q,a}$	0.8539	0.9440	0.4832
			$S_{a\|q}$	0.0045	0.3743	0.9984

Table 4: OOD question detection in QA settings. For the complete details, please see Appendix.

content. It highlights the crucial importance of adopting the likelihood ratio rather than solely relying on raw likelihood values to enhance the accuracy of OOD detection.

The Effectiveness of In-D Labels In the near OOD detection experiments presented in Table 2, Podolskiy et al. (2021)'s approach outperforms competing methods significantly. This superior performance may be attributed to the distinct class distribution across datasets. Specifically, the CLINC150 dataset comprises 150 In-D classes, in stark contrast to the SNIPS and ROSTD datasets, which offer a mere 7 (with only 5 for In-D) and 13 In-D classes, respectively. The substantially greater number of In-D classes in CLINC150 compared to the other datasets likely enhances Podolskiy et al. (2021)'s method's ability to leverage the extensive class label information, thus yielding improved OOD detection results.

Some cases that LH is better than LR In the spam detection experiments detailed in Table 3, particularly with data from the SMS and SpamAssassin datasets, we observe that without spam data, the likelihood (LH) method outperforms the likelihood ratio (LR). While LR generally shows superior and more consistent performance across most experiments — as LH can exhibit extremely poor performance in certain cases, a point elaborated in Section 6 — there are specific instances where LH is more effective.

The rationale behind using large language models (LLMs) as an OOD proxy, as introduced in Section 2, is based on the assumption that OOD data deviates from domain-specific natural language content. However, spam messages in the SMS dataset often include intentionally misspelled words to circumvent detection mechanisms, thereby violating our natural language assumption. Similarly, the content from the SpamAssassin dataset, being highly structured in email and data transaction formats (header information), also diverges from typical natural language patterns.

Given these deviations from the natural language assumption, it is understandable why LH might outperform LR in these unique scenarios.

7 RELATED WORKS

Building upon the OOD detection method using likelihood ratios introduced by Ren et al. (2019), Gangal et al. (2020) suggested employing the likelihood ratio between two LSTM language models. In their approach, one model functions as a "background model" representing OOD data and is trained on random combinations from the vocabulary.

Jin et al. (2022) used the likelihood ratio between a pretrained GPT-2 and a finetuned version of GPT-2 as a baseline to compare with their proposed contrastive learning-based OOD detection method. They showed that their proposed method can out perform the likelihood ratio based method. However, they only used GPT-2 in their likelihood ratio baseline, which is very small both in terms of training data and the model size by today standard.

In comparison, our study provides a more comprehensive analysis of likelihood ratio between base models and fine-tuned models using much larger LLMs. We show that leveraging these more advanced LLMs, likelihood ratio significantly outperforms results from Jin et al. (2022) in OOD detection. Additionally, in the current era, accessing and sharing both pretrained and finetuned LLMs has become considerably easier through online platforms such as Hugging Face. Using likelihood ratio in LLMs for OOD detection is easy, accessible, and very effective.

8 CONCLUDING REMARKS

We revisit and validate the likelihood ratio between a pretrained LLM and its finetuned variant as a criterion for OOD detection across various scenarios, without the need for additional training. This LLM-based likelihood ratio, despite being very easy to implement, shows surprising empirical effectiveness in OOD detection, and more importantly, it enables us to build robust QA systems that are able to answer both general and domain-specific questions (i.e., using the likelihood ratio to determine which LLM should be used to answer the input question). We expect that our LLM-based likelihood ratio can benefit many other applications in the future.

ACKNOWLEDGEMENTS

In the early stages of this work, AZ was supported by a personal grant from Mrs. Yanshu Wu. In the middle and later stages, AZ was funded by the UKRI AI Hub in Generative Models (EP/Y028805/1). RB and TX were supported by the German Federal Ministry of Education and Research (BMBF): Tübingen AI Center, FKZ: 01IS18039A. RB acknowledges funding by the German Research Foundation (DFG) for project 448588364 of the Emmy Noether Programme. The authors thank the International Max Planck Research School for Intelligent Systems (IMPRS-IS) for supporting Tim Z. Xiao.

References

Almeida, T. A., Hidalgo, J. M. G., and Yamakami, A. (2011). Contributions to the study of sms spam filtering: new collection and results. In *Proceedings of the 11th ACM symposium on Document engineering*, pages 259–262.

Bishop, C. M. (1994). Novelty detection and neural network validation. *IEE Proceedings-Vision, Image and Signal processing*, 141(4):217–222.

Bisk, Y., Zellers, R., Gao, J., Choi, Y., et al. (2020). Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 7432–7439.

Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., Arora, S., von Arx, S., Bernstein, M. S., Bohg, J., Bosselut, A., Brunskill, E., et al. (2021). On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*.

Borel, É. (1913). La mécanique statique et l'irréversibilité. *J. Phys. Theor. Appl.*, 3(1):189–196.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020). Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Caterini, A. L. and Loaiza-Ganem, G. (2022). Entropic issues in likelihood-based ood detection. In *I (Still) Can't Believe It's Not Better! Workshop at NeurIPS 2021*, pages 21–26. PMLR.

Cheng, D., Huang, S., and Wei, F. (2023). Adapting large language models via reading comprehension. *arXiv preprint arXiv:2309.09530*.

Clark, C., Lee, K., Chang, M.-W., Kwiatkowski, T., Collins, M., and Toutanova, K. (2019). Boolq: Exploring the surprising difficulty of natural yes/no questions. *arXiv preprint arXiv:1905.10044*.

Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., Plappert, M., Tworek, J., Hilton, J., Nakano, R., et al. (2021). Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Coucke, A., Saade, A., Ball, A., Bluche, T., Caulier, A., Leroy, D., Doumouro, C., Gisselbrecht, T., Caltagirone, F., Lavril, T., et al. (2018). Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces. *arXiv preprint arXiv:1805.10190*.

Eddington, A. (2019). *The nature of the physical world: THE GIFFORD LECTURES 1927*, volume 23. BoD–Books on Demand.

Gangal, V., Arora, A., Einolghozati, A., and Gupta, S. (2020). Likelihood ratios and generative classifiers for unsupervised out-of-domain detection in task oriented dialog. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7764–7771.

Gururangan, S., Marasović, A., Swayamdipta, S., Lo, K., Beltagy, I., Downey, D., and Smith, N. A. (2020). Don't stop pretraining: Adapt language models to domains and tasks. *arXiv preprint arXiv:2004.10964*.

Hendrycks, D., Burns, C., Kadavath, S., Arora, A., Basart, S., Tang, E., Song, D., and Steinhardt, J. (2021). Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*.

Hendrycks, D. and Gimpel, K. (2016). A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136*.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. (2021). Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Jiang, A. Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D. S., Casas, D. d. l., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., et al. (2023). Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Jin, D., Gao, S., Kim, S., Liu, Y., and Hakkani-Tür, D. (2022). Towards textual out-of-domain detection without in-domain labels. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30:1386–1395.

Jin, Q., Dhingra, B., Liu, Z., Cohen, W. W., and Lu, X. (2019). Pubmedqa: A dataset for biomedical research question answering. *arXiv preprint arXiv:1909.06146*.

Khosla, P., Teterwak, P., Wang, C., Sarna, A., Tian, Y., Isola, P., Maschinot, A., Liu, C., and Krishnan, D. (2020). Supervised contrastive learning. *Advances in neural information processing systems*, 33:18661–18673.

Labonne, M. and Moran, S. (2023). Spam-t5: Benchmarking large language models for few-shot email spam detection. *arXiv preprint arXiv:2304.01238*.

Lang, K. (1995). Newsweeder: Learning to filter netnews. In *Machine learning proceedings 1995*, pages 331–339. Elsevier.

Larson, S., Mahendran, A., Peper, J. J., Clarke, C., Lee, A., Hill, P., Kummerfeld, J. K., Leach, K., Laurenzano, M. A., Tang, L., et al. (2019). An evaluation dataset for intent classification and out-of-scope prediction. *arXiv preprint arXiv:1909.02027*.

Lee, K., Lee, K., Lee, H., and Shin, J. (2018). A simple unified framework for detecting out-of-distribution samples and adversarial attacks. *Advances in neural information processing systems*, 31.

Liu, W., Wang, X., Owens, J., and Li, Y. (2020). Energy-based out-of-distribution detection. *Advances in neural information processing systems*, 33:21464–21475.

Lyu, Z., Duolikun, D., Dai, B., Yao, Y., Minervini, P., Xiao, T. Z., and Gal, Y. (2020). You need only uncertain answers: Data efficient multilingual question answering. *ICML 2020 Workshop on Uncertainty and Robustness in Deep Learning*.

Metsis, V., Androutsopoulos, I., and Paliouras, G. (2006). Spam filtering with naive bayes-which naive bayes? In *CEAS*, volume 17, pages 28–69. Mountain View, CA.

Nalisnick, E., Matsukawa, A., Teh, Y. W., Gorur, D., and Lakshminarayanan, B. (2018). Do deep generative models know what they don't know? *arXiv preprint arXiv:1810.09136*.

Podolskiy, A., Lipin, D., Bout, A., Artemova, E., and Piontkovskaya, I. (2021). Revisiting mahalanobis distance for transformer-based out-of-domain detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 13675–13682.

Rajpurkar, P., Jia, R., and Liang, P. (2018). Know what you don't know: Unanswerable questions for squad. *arXiv preprint arXiv:1806.03822*.

Ren, J., Liu, P. J., Fertig, E., Snoek, J., Poplin, R., Depristo, M., Dillon, J., and Lakshminarayanan, B. (2019). Likelihood ratios for out-of-distribution detection. *Advances in neural information processing systems*, 32.

Sakkis, G., Androutsopoulos, I., Paliouras, G., Karkaletsis, V., Spyropoulos, C. D., and Stamatopoulos, P. (2003). A memory-based approach to anti-spam filtering for mailing lists. *Information retrieval*, 6:49–73.

Schirrmeister, R., Zhou, Y., Ball, T., and Zhang, D. (2020). Understanding anomaly detection with deep invertible networks through hierarchies of distributions and features. *Advances in Neural Information Processing Systems*, 33:21038–21049.

Serrà, J., Álvarez, D., Gómez, V., Slizovskaia, O., Núñez, J. F., and Luque, J. (2019). Input complexity and out-of-distribution detection with likelihood-based generative models. *arXiv preprint arXiv:1909.11480*.

Sun, Y., Ming, Y., Zhu, X., and Li, Y. (2022). Out-of-distribution detection with deep nearest neighbors. In *International Conference on Machine Learning*, pages 20827–20840. PMLR.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al. (2023). Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Uppaal, R., Hu, J., and Li, Y. (2023). Is fine-tuning needed? pre-trained language models are near perfect for out-of-domain detection. *arXiv preprint arXiv:2305.13282*.

Wang, H., Li, Z., Feng, L., and Zhang, W. (2022). Vim: Out-of-distribution with virtual-logit matching. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4921–4930.

Xiao, T. Z., Gomez, A. N., and Gal, Y. (2020). Wat zei je? detecting out-of-distribution translations with variational transformers. *arXiv preprint arXiv:2006.08344*.

Yang, J., Wang, P., Zou, D., Zhou, Z., Ding, K., Peng, W., Wang, H., Chen, G., Li, B., Sun, Y., et al. (2022). Openood: Benchmarking generalized out-of-distribution detection. *Advances in Neural Information Processing Systems*, 35:32598–32611.

Yu, L., Jiang, W., Shi, H., Yu, J., Liu, Z., Zhang, Y., Kwok, J. T., Li, Z., Weller, A., and Liu, W. (2023). Metamath: Bootstrap your own mathematical questions for large language models. *arXiv preprint arXiv:2309.12284*.

Zhang, A. and Wischik, D. (2022). Falsehoods that ml researchers believe about ood detection. *arXiv preprint arXiv:2210.12767*.

Zhang, M., Zhang, A., and McDonagh, S. (2021). On the out-of-distribution generalization of probabilistic image modelling. *Advances in Neural Information Processing Systems*, 34:3811–3823.

Zheng, L., Guha, N., Anderson, B. R., Henderson, P., and Ho, D. E. (2021). When does pretraining help? assessing self-supervised learning for law and the casehold dataset of 53,000+ legal holdings. In *Proceedings of the eighteenth international conference on artificial intelligence and law*, pages 159–168.

Zhou, W., Liu, F., and Chen, M. (2021). Contrastive out-of-distribution detection for pretrained transformers. *arXiv preprint arXiv:2104.08812*.

Zhu, Y., Chen, Y., Xie, C., Li, X., Zhang, R., Xue, H., Tian, X., Chen, Y., et al. (2022). Boosting out-of-distribution detection with typical features. *Advances in Neural Information Processing Systems*, 35:20758–20769.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. Yes
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. Not Applicable
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. Yes

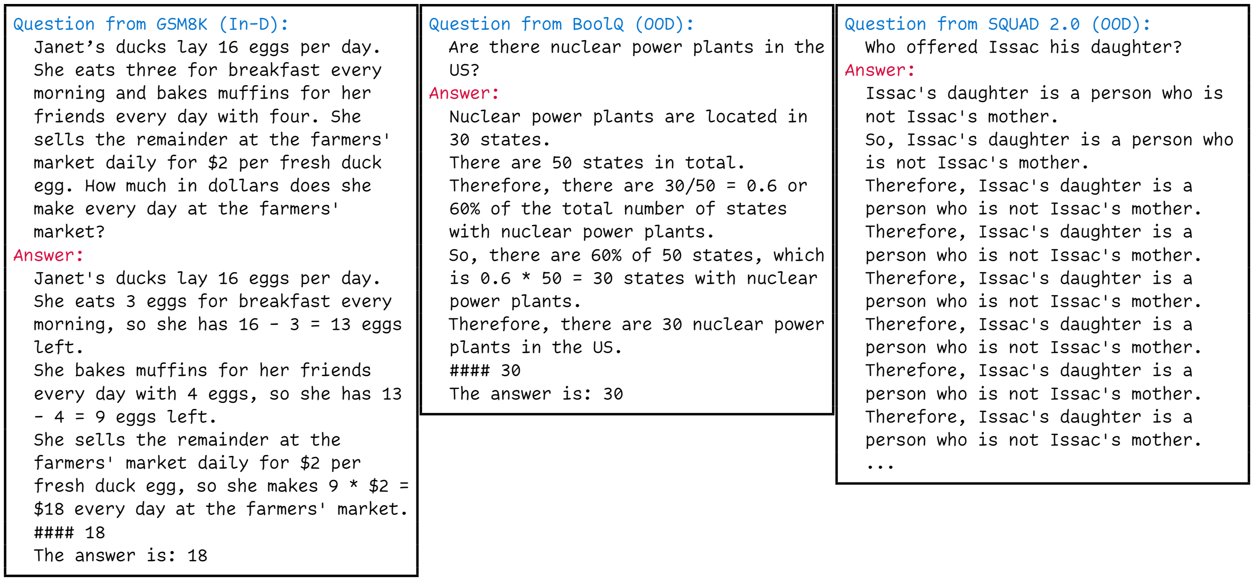
2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. Not Applicable
   (b) Complete proofs of all theoretical results. Not Applicable
   (c) Clear explanations of any assumptions. Not Applicable

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). Yes
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). Yes
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). Yes
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). Yes

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. Yes
   (b) The license information of the assets, if applicable. Yes
   (c) New assets either in the supplemental material or as a URL, if applicable. Not Applicable
   (d) Information about consent from data providers/curators. Not Applicable
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. Not Applicable

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. Not Applicable
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. Not Applicable
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. Not Applicable

Your Finetuned Large Language Model is Already a Powerful Out-of-distribution Detector: Supplementary Materials



```
Question from GSM8K (In-D):
  Janet’s ducks lay 16 eggs per day.
  She eats three for breakfast every
  morning and bakes muffins for her
  friends every day with four. She
  sells the remainder at the farmers'
  market daily for $2 per fresh duck
  egg. How much in dollars does she
  make every day at the farmers'
  market?
Answer:
  Janet's ducks lay 16 eggs per day.
  She eats 3 eggs for breakfast every
  morning, so she has 16 - 3 = 13 eggs
  left.
  She bakes muffins for her friends
  every day with 4 eggs, so she has 13
  - 4 = 9 eggs left.
  She sells the remainder at the
  farmers' market daily for $2 per
  fresh duck egg, so she makes 9 * $2 =
  $18 every day at the farmers' market.
  #### 18
  The answer is: 18
```

```
Question from BoolQ (OOD):
  Are there nuclear power plants in the
  US?
Answer:
  Nuclear power plants are located in
  30 states.
  There are 50 states in total.
  Therefore, there are 30/50 = 0.6 or
  60% of the total number of states
  with nuclear power plants.
  So, there are 60% of 50 states, which
  is 0.6 * 50 = 30 states with nuclear
  power plants.
  Therefore, there are 30 nuclear power
  plants in the US.
  #### 30
  The answer is: 30
```

```
Question from SQUAD 2.0 (OOD):
  Who offered Issac his daughter?
Answer:
  Issac's daughter is a person who is
  not Issac's mother.
  So, Issac's daughter is a person who
  is not Issac's mother.
  Therefore, Issac's daughter is a
  person who is not Issac's mother.
  Therefore, Issac's daughter is a
  person who is not Issac's mother.
  Therefore, Issac's daughter is a
  person who is not Issac's mother.
  Therefore, Issac's daughter is a
  person who is not Issac's mother.
  Therefore, Issac's daughter is a
  person who is not Issac's mother.
  Therefore, Issac's daughter is a
  person who is not Issac's mother.
  ...
```

Figure 3: Example question-answer sets produced by MetaMath-7B. The responses to In-D questions are accurate and logical. However, for OOD questions, MetaMath-7B generates unreasonable answers, responding to a straightforward query with unnecessary mathematical calculations or producing repetitive sentences with no useful information.

9 Finetuning Details

For finetuning, with the exception of MetaMath (Yu et al., 2023), which is comprehensively finetuned from the Llama-2 model, we employ LoRA (Low-Rank Adaptation) as described by Hu et al. (2021) for adjusting the models. Our aim is to demonstrate the general applicability of our method; therefore, all our LoRA finetuning follows a uniform parameter configuration. In accordance with prevalent practices, we finetune the parameters in the Q (query) and V (value) projections. We employ a LoRA rank of 16, a LoRA alpha of 32, and a LoRA dropout rate of 0.05. Given the varying sizes of datasets, we adjust the learning rate between 10^{-3} and 10^{-4}, and the training typically spans 10 to 20 epochs. We select the model checkpoint with the best evaluation loss as our finetuned model. For detailed information, please refer to the code provided.

10 The Use of AI Assistants in Research and Writing

We assert that our use of AI assistants is strictly limited to revising the original text.

11 Description of Computing Infrastructure

All experiments were conducted using NVIDIA GPUs with varying memory capacities based on model size requirements. For models with parameter counts below 7B, we utilized NVIDIA V100 GPUs with 32GB VRAM and NVIDIA RTX 4090 GPUs with 24GB VRAM. For larger models with 13B parameters, we employed NVIDIA H100 GPUs with 80GB VRAM to accommodate the increased memory demands during training and inference.

12 Full Experiment Results

Full experiment results are detailed below. All the results are averaged over five runs.

In-D	OOD	Method	AUROC ↑	AUPR (OOD) ↑	FPR95 ↓
20NG	SST-2	Zhou et al. (2021)	0.978	0.865	0.015
		CE (Hendrycks and Gimpel, 2016)	0.981	0.942	0.087
		TAPT (Gururangan et al., 2020)	0.981	0.939	0.088
		SupCon (Khosla et al., 2020)	0.980	0.943	0.094
		Uppaal et al. (2023)	**1.000**	0.999	**0.000**
		Llama-7B LH	0.008	0.541	0.999
		Llama-7B LR	**1.000**	**1.000**	**0.000**
		Mistral-7B LH	0.008	0.541	1.000
		Mistral-7B LR	0.995	0.999	0.009
		Llama-13B LH	0.009	0.541	1.000
		Llama-13B LR	**1.000**	**1.000**	**0.000**
	MNLI	Zhou et al. (2021)	0.964	0.978	0.224
		CE (Hendrycks and Gimpel, 2016)	0.968	0.989	0.166
		TAPT (Gururangan et al., 2020)	0.964	0.988	0.175
		SupCon (Khosla et al., 2020)	0.970	0.990	0.156
		Uppaal et al. (2023)	**1.000**	**1.000**	**0.000**
		Llama-7B LH	0.020	0.119	0.999
		Llama-7B LR	**1.000**	**1.000**	0.001
		Mistral-7B LH	0.024	0.119	0.999
		Mistral-7B LR	0.996	0.996	0.008
		Llama-13B LH	0.024	0.119	0.998
		Llama-13B LR	**1.000**	**1.000**	**0.000**
	RTE	Zhou et al. (2021)	0.956	0.860	0.312
		CE (Hendrycks and Gimpel, 2016)	0.945	0.902	0.285
		TAPT (Gururangan et al., 2020)	0.919	0.869	0.352
		SupCon (Khosla et al., 2020)	0.952	0.914	0.248
		Uppaal et al. (2023)	**1.000**	0.999	**0.000**
		Llama-7B LH	0.063	0.443	0.998
		Llama-7B LR	**1.000**	**1.000**	0.001
		Mistral-7B LH	0.074	0.446	0.998
		Mistral-7B LR	0.997	0.999	0.006
		Llama-13B LH	0.070	0.445	0.997
		Llama-13B LR	**1.000**	**1.000**	**0.000**
	IMDB	Zhou et al. (2021)	0.969	0.996	0.144
		CE (Hendrycks and Gimpel, 2016)	0.961	0.995	0.206
		TAPT (Gururangan et al., 2020)	0.965	0.995	0.159
		SupCon (Khosla et al., 2020)	0.970	0.996	0.150
		Uppaal et al. (2023)	0.990	0.998	0.012
		Llama-7B LH	0.755	0.311	0.932
		Llama-7B LR	**1.000**	**1.000**	0.001
		Mistral-7B LH	0.767	0.943	0.926
		Mistral-7B LR	0.999	0.998	0.003
		Llama-13B LH	0.773	0.332	0.919
		Llama-13B LR	**1.000**	**1.000**	**0.000**
	Multi30K	Zhou et al. (2021)	0.980	0.888	0.005
		CE (Hendrycks and Gimpel, 2016)	0.962	0.920	0.175
		TAPT (Gururangan et al., 2020)	0.956	0.922	0.167
		SupCon (Khosla et al., 2020)	0.955	0.918	0.201
		Uppaal et al. (2023)	**1.000**	**1.000**	**0.000**
		Llama-7B LH	0.002	0.470	1.000
		Llama-7B LR	**1.000**	**1.000**	**0.000**
		Mistral-7B LH	0.002	0.470	1.000
		Mistral-7B LR	0.995	0.998	0.008
		Llama-13B LH	0.002	0.470	1.000
		Llama-13B LR	**1.000**	**1.000**	**0.000**
	NewsCategory	Zhou et al. (2021)	0.955	0.969	0.383
		CE (Hendrycks and Gimpel, 2016)	0.957	0.984	0.234
		TAPT (Gururangan et al., 2020)	0.947	0.981	0.243
		SupCon (Khosla et al., 2020)	0.962	0.986	0.219
		Uppaal et al. (2023)	**1.000**	**1.000**	**0.000**
		Llama-7B LH	0.014	0.128	1.000
		Llama-7B LR	**1.000**	**1.000**	0.001
		Mistral-7B LH	0.019	0.129	1.000
		Mistral-7B LR	0.997	0.997	0.006
		Llama-13B LH	0.017	0.128	0.999
		Llama-13B LR	**1.000**	**1.000**	**0.000**
	CLINC150	Zhou et al. (2021)	0.988	0.870	0.005
		CE (Hendrycks and Gimpel, 2016)	0.964	0.844	0.189
		TAPT (Gururangan et al., 2020)	0.959	0.830	0.213
		SupCon (Khosla et al., 2020)	0.957	0.821	0.230
		Uppaal et al. (2023)	**1.000**	**1.000**	**0.000**
		Llama-7B LH	0.001	0.661	1.000
		Llama-7B LR	**1.000**	**1.000**	**0.000**
		Mistral-7B LH	0.001	0.661	1.000
		Mistral-7B LR	0.995	**1.000**	0.008
		Llama-13B LH	0.001	0.661	1.000
		Llama-13B LR	**1.000**	**1.000**	**0.000**

Table 5: Results of far OOD detection, utilizing the same experimental setup as described by Uppaal et al. (2023). Results for methods not originating from our work are cited directly from Uppaal et al. (2023).

Dataset	Spam Data	Model	AUROC ↑	AUPR ↑	FPR95 ↓
Ling	No	Llama-7B LH	0.552	0.215	0.934
		Llama-7B LR	0.967	0.858	0.174
		Llama-13B LH	0.534	0.182	0.929
		Llama-13B LR	0.933	0.746	0.336
	Yes	NB	1.000	1.000	0.000
		Logistic	1.000	1.000	0.000
		KNN	0.968	0.932	0.021
		SVM	1.000	1.000	0.000
		XGBoost	0.995	0.973	0.017
		LightGBM	0.997	0.979	0.008
		RoBERTa	1.000	1.000	0.000
		Spam-T5	1.000	1.000	0.000
		Llama-7B LR	0.998	0.993	0.008
		Llama-13B LR	0.997	0.988	0.012
SMS	No	Llama-7B LH	0.960	0.699	0.088
		Llama-7B LR	0.866	0.582	0.487
		Llama-13B LH	0.957	0.689	0.093
		Llama-13B LR	0.810	0.518	0.761
	Yes	NB	0.988	0.949	0.113
		Logistic	0.985	0.946	0.124
		KNN	0.863	0.830	0.811
		SVM	0.997	0.980	0.024
		XGBoost	0.918	0.873	0.676
		LightGBM	0.978	0.921	0.103
		RoBERTa	0.997	0.988	0.004
		Spam-T5	0.985	0.959	0.082
		Llama-7B LR	1.000	1.000	0.000
		Llama-13B LR	0.999	0.995	0.000
SpamAssassin	No	Llama-7B LH	0.964	0.884	0.096
		Llama-7B LR	0.960	0.935	0.296
		Llama-13B LH	0.956	0.897	0.169
		Llama-13B LR	0.941	0.917	0.398
	Yes	NB	0.971	0.917	0.070
		Logistic	0.992	0.986	0.029
		KNN	0.931	0.935	0.578
		SVM	0.990	0.983	0.046
		XGBoost	0.994	0.989	0.019
		LightGBM	1.000	0.999	0.000
		RoBERTa	0.999	0.997	0.000
		Spam-T5	0.996	0.994	0.012
		Llama-7B LR	0.998	0.996	0.005
		Llama-13B LR	0.994	0.989	0.019
Enron	No	Llama-7B LH	0.721	0.728	0.798
		Llama-7B LR	0.991	0.989	0.043
		Llama-13B LH	0.719	0.723	0.798
		Llama-13B LR	0.992	0.990	0.035
	Yes	NB	0.992	0.991	0.035
		Logistic	0.994	0.992	0.025
		KNN	0.915	0.927	0.239
		SVM	0.998	0.998	0.008
		XGBoost	0.975	0.967	0.111
		LightGBM	0.997	0.997	0.013
		RoBERTa	1.000	1.000	0.001
		Spam-T5	1.000	1.000	0.001
		Llama-7B LR	0.999	0.999	0.001
		Llama-13B LR	1.000	1.000	0.000

Table 6: Results of spam detection.

In-D	OOD	Model	Criterion	AUROC ↑	AUPR (OOD) ↑	FPR95 ↓
GSM8K	SQUAD 2.0	MetaMath-7B	S_q	0.1116	0.0546	0.9894
			S_a	0.5463	0.0979	0.9947
			$S_{q,a}$	0.5363	0.0959	0.9924
			$S_{a\|q}$	**0.6877**	**0.1376**	**0.9704**
		MetaMath-13B	S_q	0.0519	0.0524	0.9955
			S_a	0.4958	0.0891	0.9955
			$S_{q,a}$	0.4017	0.0764	0.9947
			$S_{a\|q}$	**0.6144**	**0.1136**	**0.9765**
	BoolQ	MetaMath-7B	S_q	0.0538	0.1618	0.9955
			S_a	0.5045	0.2616	0.9932
			$S_{q,a}$	0.4797	0.2536	0.9879
			$S_{a\|q}$	**0.7156**	**0.4041**	**0.9310**
		MetaMath-13B	S_q	0.1008	0.1659	0.9924
			S_a	0.5507	0.2861	0.9833
			$S_{q,a}$	0.4778	0.2530	0.9795
			$S_{a\|q}$	**0.6967**	**0.3886**	**0.9303**
	PIQA	MetaMath-7B	S_q	0.9762	0.9779	0.0735
			S_a	0.9612	0.9746	0.0569
			$S_{q,a}$	**0.9975**	**0.9983**	**0.0038**
			$S_{a\|q}$	0.9944	0.9944	0.0099
		MetaMath-13B	S_q	0.9900	0.9906	0.0318
			S_a	0.8879	0.9331	0.1122
			$S_{q,a}$	**1.0000**	0.9999	**0.0000**
			$S_{a\|q}$	**1.0000**	**1.0000**	**0.0000**
MATH	SQUAD 2.0	MetaMath-7B	S_q	0.2139	0.2304	0.9474
			S_a	0.6384	0.3963	0.8916
			$S_{q,a}$	0.6527	0.4477	0.8436
			$S_{a\|q}$	**0.7385**	**0.5305**	**0.7914**
		MetaMath-13B	S_q	0.1880	0.2232	0.9460
			S_a	0.6098	0.3841	0.8890
			$S_{q,a}$	0.5628	0.3649	0.8882
			$S_{a\|q}$	**0.6786**	**0.4737**	**0.8166**
	BoolQ	MetaMath-7B	S_q	0.1303	0.4472	0.9658
			S_a	0.6135	0.7111	0.8580
			$S_{q,a}$	0.6361	0.7474	0.8008
			$S_{a\|q}$	**0.7507**	**0.8266**	**0.6870**
		MetaMath-13B	S_q	0.2612	0.5223	0.9304
			S_a	0.6488	0.7474	0.8148
			$S_{q,a}$	0.6384	0.7521	0.7818
			$S_{a\|q}$	**0.7350**	**0.8191**	**0.6854**
	PIQA	MetaMath-7B	S_q	0.9681	0.9902	0.0732
			S_a	0.9206	0.9775	0.1133
			$S_{q,a}$	0.9873	**0.9962**	**0.0242**
			$S_{a\|q}$	**0.9876**	0.9956	0.0812
		MetaMath-13B	S_q	0.9795	0.9938	0.0452
			S_a	0.8495	0.9572	0.1627
			$S_{q,a}$	**0.9897**	**0.9969**	**0.0198**
			$S_{a\|q}$	0.9626	0.9895	0.0484

Table 7: Outcomes of OOD question detection in QA settings. This table continues in Table 8.

In-D	**OOD**	**Model**	**Criterion**	**AUROC ↑**	**AUPR (OOD) ↑**	**FPR95 ↓**
casehold	SQUAD 2.0	law-chat-7B	S_q	0.2463	0.2039	0.9987
			S_a	**0.9082**	**0.8425**	**0.3717**
			$S_{q,a}$	0.2048	0.1954	0.9989
			$S_{a\|q}$	0.3915	0.2080	0.9992
	BoolQ	law-chat-7B	S_q	0.3869	0.5249	0.9985
			S_a	**0.8878**	**0.9222**	**0.4753**
			$S_{q,a}$	0.2692	0.4737	0.9988
			$S_{a\|q}$	0.1307	0.3257	0.9996
	PIQA	law-chat-7B	S_q	0.9241	0.9717	0.3064
			S_a	**0.9917**	**0.9978**	**0.0078**
			$S_{q,a}$	0.8539	0.9440	0.4832
			$S_{a\|q}$	0.0045	0.3743	0.9984
PubMedQA	SQUAD 2.0	medicine-chat-7B	S_q	**0.9024**	0.2729	0.4300
			S_a	0.7939	**0.6476**	**0.3660**
			$S_{q,a}$	0.8967	0.3959	0.4360
			$S_{a\|q}$	0.5756	0.1869	0.8400
	BoolQ	medicine-chat-7B	S_q	**0.9266**	0.5906	0.3980
			S_a	0.8009	**0.7057**	**0.3800**
			$S_{q,a}$	0.9066	0.6281	0.4440
			$S_{a\|q}$	0.3515	0.2356	0.8440
	PIQA	medicine-chat-7B	S_q	**0.9998**	**0.9994**	**0.0000**
			S_a	0.8606	0.8439	0.3000
			$S_{q,a}$	0.9995	0.9984	0.0020
			$S_{a\|q}$	0.2471	0.2892	0.8420

Table 8: Outcomes of OOD question detection in QA settings (continued).