SHH, DON'T SAY THAT! DOMAIN CERTIFICATION IN LLMS

Cornelius Emde[1]*, **Preetham Arvind**[1], **Alasdair Paren**[1], **Maxime Kayser**[1], **Tom Rainforth**[1]
Thomas Lukasiewicz[2], **Philip Torr**[1], **Adel Bibi**[1]
[1]University of Oxford [2]Vienna University of Technology

ABSTRACT

Large language models (LLMs) are often deployed to perform constrained tasks, with narrow domains. For example, customer support bots can be built on top of LLMs, relying on their broad language understanding and capabilities to enhance performance. However, these LLMs are adversarially susceptible, potentially generating outputs outside the intended domain. To formalize, assess and mitigate this risk, we introduce *domain certification*; a guarantee that accurately characterizes the out-of-domain behaviour of language models. We then propose a simple yet effective approach which we call VALID that provides adversarial bounds as a certificate. Finally, we evaluate our method across a diverse set of datasets, demonstrating that it yields meaningful certificates.

1 INTRODUCTION

With recent advancements in the field of natural language processing, large language models (LLMs) have become ubiquitous. In particular, the scaling of recent large generalist models dubbed foundation models has shown to possess emergent abilities that benefit a wide range of downstream tasks such as text generation, question answering and text comprehension (Kaplan et al., 2020; Alabdulmohsin et al., 2022; Xiong et al., 2024; Henighan et al., 2020; Brown et al., 2020). Adapting these foundation models for downstream tasks often leads to the state-of-the-art performance and has become the dominant paradigm (Gao et al., 2020). This is typically achieved via fine-tuning on task-relevant data (e.g. via Low-Rank Adaptation (LoRA) Hu et al. (2021), in-context learning (Mosbach et al., 2023), prefix tuning (Li & Liang, 2021), or simply prompt engineering.

However, foundation models are typically trained on large amounts of unlabeled web data which contains a wide range of information that is either irrelevant to a task or potentially harmful (Bommasani et al., 2022). Therefore, it is desirable to restrict the output of a generalist LLM to a specific domain. For instance, consider the government of *Atlantis* providing a general purpose chatbot to advise their citizens on tax laws and support them in doing their tax reports. It would be important, for public reputation and cost reasons that such a system would remain on topic and could not be misused, either intentionally or unintentionally.

While deployers commonly want LLMs to only give responses on a certain set of topics, an adversary might try to elicit an arbitrary response. Thus, a successful attack is an input string that creates a coherent response outside the target domain. There are various reasons why an adversary might want to elicit such a response that is out-of-domain (OOD). The adversarial user might want to misappropriate the system as a cost-effective tool for a purpose it wasn't built for, resulting in excess infrastructure costs for the deployer. There have been cases where, for instance, a shopping chatbot has been misused to write code (Kishan, 2023). Conversely, the deployer might legally be required to validate and verify their models, which is challenging, if not impossible, when the model is not domain-restricted. Finally, the adversary might want to harm the company directly by eliciting harmful OOD responses, which could damage the company's reputation when publicised. Such examples have become more common recently (McClure, 2023; The Guardian, 2023). Deployers have moral and legal obligations to prevent this (Bommasani et al., 2022). In all examples, restricting the domain in which the model responds *under adversarial* prompts can help mitigate risks. Thus, in the era of foundation models, "domain" specialization is critical.

Existing work has implemented guardrails that address these risks (Jain et al., 2023), most notably via alignment, resulting in the models to reject user requests (Bai et al., 2022; Ouyang et al., 2022;

[*] Corresponding Author `cornelius.emde[AT]cs[DOT]ox[DOT]ac[DOT]uk`

Christiano et al., 2023). However, a wide body of research has shown that such guardrails have "jailbreaks", i.e., they can easily be circumvented by a motivated adversary (Wang et al., 2024; Qi et al., 2024; Carlini et al., 2024b; Dong et al., 2024a). Common jailbreak methods are prompt injection (Perez & Ribeiro, 2022; Jiang et al., 2023; Liu et al., 2024), numerical optimization (Jia & Liang, 2017; Wallace et al., 2021a; Ebrahimi et al., 2018; Jones et al., 2023; Zou et al., 2023; Jia et al., 2024), red teaming (Perez et al., 2022; Samvelyan et al., 2024), automated black-box attacks (Chao et al., 2024; Mehrotra et al., 2024), or data poisoning attacks (Biggio et al., 2013; Wallace et al., 2021b; Carlini et al., 2024a). Using these tools, it is possible for adversaries to retrieve information from a fine-tuned model that was suppressed by the alignment and generate responses that are outside the target domain. Adversarial prefixes or suffixes that augment any prompt are especially powerful as they have been shown to universally attack models in combination with a wide range of prompts and can thus be shared between adversarial users (Wallace et al., 2021a; Zou et al., 2023). This presents a significant risk. Hence, researchers have proposed methods to defend against these adversarial attacks, such as unlearning (Nguyen et al., 2022; Xu et al., 2023), robust fine-tuning (O'Neill et al., 2023; Dong et al., 2021), or request and response filtering (Inan et al., 2023a). We provide a more structured review of relevant literature in Appendix F.

Deployers would ideally want guardrails that come with a provable, mathematical guarantee against the model responding off topic, or a guarantee that it does this with very low probability. The process of constructing guarantees against certain model behaviours under adversarial attack is commonly referred to as *certification* and has been successfully applied to vision applications in recent years (Akhtar et al., 2021). However, no existing LLM guardrails provide guaranteed protection against existing or future jailbreaking techniques, leaving deployed models at risk of being compromised shortly after release. As a result, developing certifiable methods to guarantee that specialized LLMs consistently produce on topic content is critical. To that end, our contributions are as follows:

- We introduce a novel framework, *domain certification*, to bound the probability of models producing out-of-domain content under adversarial attack.
- We introduce an easy-to-use algorithm VALID that bounds the probability of an LLM based system responding off topic under adversarial attack. We show the efficiency of VALID which we test empirically on a number of representative data sets.

2 Domain Certification

We now introduce our *domain-certification* framework for offering mathematical guarantees that a LLM system stays on topic. In Section 2.1, we formally introduce this framework. In Section 3, we present **V**erified **A**dversarial **L**LM Output via **I**terative **D**ismissal (VALID). VALID is an easy-to-use method to create a system that adheres to these guarantees. In plain language, we are proposing a certifiable guardrail for LLM-driven systems as follows:

> *A model is* domain-certified*, when an adversarial upper bound can be placed on the probability that the model provides an output outside its designated target domain.*

Before formalizing this statement, we introduce some mathematical notation. We represent tokens (i.e. individual text units), as x and y, which belong to the token space $x, y \in \mathbb{V}$ where $\mathbb{V} = \{1, \dots, V\}$ (a vocabulary of size V). We define the space of sequences of arbitrary length as $\mathbb{S} \triangleq \mathbb{V}*$, where $*$ symbolizes the Kleene closure. Sequences of tokens are denoted by bold letters with $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{S}$, with $\boldsymbol{x}$ and $\boldsymbol{y}$ representing the input and output sequences of a LLM respectively. We use lowercase letters to denote models that predict the next token, such as $l : \mathbb{S} \rightarrow \mathbb{V}$. Applying this model repeatedly, until the end-of-sequence token creates a sequence-to-sequence model $L : \mathbb{S} \rightarrow \mathbb{S}$. We denote the likelihood of sample $\boldsymbol{y}$ under L given $\boldsymbol{x}$ as $L(\boldsymbol{y}|\boldsymbol{x})$, which is obtained by $L(\boldsymbol{y}|\boldsymbol{x}) = \prod_{n=1}^{N_y} l(y_n|y_{<n}, \boldsymbol{x})$ for a sentence $\boldsymbol{y}$ of length N_y. We further denote the distribution from which the model samples its output as $\boldsymbol{y} \sim L(\cdot|\boldsymbol{x})$.

2.1 Defining Domain Certification

We now formally introduce *domain certification*. We define the target domain (or set of desired topics) as a subset of the sentence space $\mathbb{S}$. We partition $\mathbb{S}$ into the target domain $\mathbb{T}$ and its compliment $\mathbb{T}'$. For instance, $\mathbb{T}$ might be all sentences in $\mathbb{S}$ meaningfully occurring for "question answering for tax reports". In addition, we define the set of unwanted responses as $\mathbb{F} \subset \mathbb{T}'$ ($\mathbb{F}$ as "forbidden"). In the following we will certify with respect to this set $\mathbb{F}$ rather than $\mathbb{T}$. We choose $\mathbb{F}$ such that sequences in $\mathbb{F}' \cap \mathbb{T}'$ do not pose a risk such as described in Section 1. For instance, they might be unintelligible or meaningless sequences of tokens. Hence, we wish to establish a guarantee that L is unlikely to produce an output in $\mathbb{F}$. First, we define a bound for a given element $\boldsymbol{y}$ in $\mathbb{S}$:

Definition 1 ***Atomic Certificate***. *We say a model $L : \mathbb{S} \to \mathbb{S}$ is $\epsilon_{\boldsymbol{y}}$-atomic-certified ($\epsilon_{\boldsymbol{y}}$-AC) for some sample $\boldsymbol{y}$ (i.e. an* atom*) in the output set $\mathbb{S}$, iff*

$$\forall \boldsymbol{x} \in \mathbb{S} : L(\boldsymbol{y}|\boldsymbol{x}) \leq \epsilon_{\boldsymbol{y}}. \tag{1}$$

In words, a model that is $\epsilon_{\boldsymbol{y}}$-AC for a sample $\boldsymbol{y}$, will generate sample $\boldsymbol{y}$ with probability smaller than $\epsilon_{\boldsymbol{y}}$ for *any* $\boldsymbol{x} \in \mathbb{S}$, and hence for adversarially chosen $\boldsymbol{x}$. If this is the case, we say model L is *certifiable* for sample $\boldsymbol{y}$ with $\epsilon_{\boldsymbol{y}}$, i.e. $\epsilon_{\boldsymbol{y}}$ is the *smallest* value that probably bounds L. Ideally, such an upper bound $\epsilon_{\boldsymbol{y}}$ would be large for samples in the target domain $\mathbb{T}$, meaning the certificate is permissive, and small for sample drawn from $\mathbb{F}$ meaning the certificate is restrictive. Having obtained the ACs for a set of undesirable responses, we can translate these ACs into one certificate across the entire set $\mathbb{F}$ by studying the worst-case across $\mathbb{F}$:

Definition 2 ***Domain Certificate***. *We say model L is ϵ-domain-certified (ϵ-DC) with respect to $\mathbb{F}$, when it is $\epsilon_{\boldsymbol{y}}$-AC for all $\boldsymbol{y} \in \mathbb{F}$ with $\epsilon_{\boldsymbol{y}} \leq \epsilon$:*

$$\forall \boldsymbol{x} \in \mathbb{S}, \boldsymbol{y} \in \mathbb{F} : L(\boldsymbol{y}|\boldsymbol{x}) \leq \epsilon. \tag{2}$$

This imposes a global bound on L across all undesired responses in $\mathbb{F}$. In practice, we cannot establish the ϵ-DC certificate w.r.t. $\mathbb{F}$ as we cannot enumerate $\mathbb{F}$. Hence, following standard practice in ML evaluation we propose to use $\mathcal{D}_{\mathbb{F}}$, a finite dataset of responses in $\mathbb{F}$ to establish a ϵ-DC certificate w.r.t. $\mathcal{D}_{\mathbb{F}}$ approximating the certificate for $\mathbb{F}$.

Recent discussions have raised the need for bounds on undesirable behaviour. For instance, Yoshua Bengio has advocated for upper bounds on harmful behaviour (Bengio et al., 2024) in a recent blog post (Bengio, 2024a). In addition, an increasing body of legislation mandates thorough auditing of ML systems EU (2024). The atomic and domain certificates can play a vital role in assessing the risk of worst-case behaviour. For instance, consider the deployer of a LLM-based system that processes 10 requests per second. The deployer might perform an a priori risk assessment and determine that they can tolerate the consequences of an out-of-domain response from $\mathcal{D}_{\mathbb{F}}$ per year. The deployer should certify the LLM system as ϵ-DC with $\epsilon \approx 10^{-8}$ in order to achieve this level of risk.

3 ACHIEVING DOMAIN CERTIFICATION

In this section, we introduce **V**erified **A**dversarial **LL**M Output via **I**terative **D**ismissal (VALID) to obtain certification as described in Definitions 1 and 2.

Algorithm 1 VALID

```
Require: LLM $L$, Guide model $G$, hyperparameters $k$ and $T$, prompt $\boldsymbol{x}$
  for $t \in \{1, \ldots, T\}$ do
      Sample $\boldsymbol{y} \sim L(\cdot|\boldsymbol{x})$
      $N_{\boldsymbol{y}} \leftarrow$ length($\boldsymbol{y}$)
      if $\log \frac{L(\boldsymbol{y}|\boldsymbol{x})}{G(\boldsymbol{y})} \leq kN_{\boldsymbol{y}}$ then
          Return: $\boldsymbol{y}$
  Return: "Abstained".
```

In order to motivate VALID we first discuss how one might generate certificates with the simplifying assumption of having assess to a domain oracle Ω. Specifically, we define Ω as a *generator* for domain $\mathbb{T}$: Ω assigns high likelihood to sentences in $\mathbb{T}$ and zero likelihood to elements in $\mathbb{F}$. In this case one could simply bound the divergence between L and Ω to restrict the model domain. In particular, we might use the Renyi divergence of order infinity, $\Delta_\infty(P \parallel Q) \triangleq \log \sup_x \frac{P(x)}{Q(x)}$ (Rényi, 1961).

$$\forall \boldsymbol{x} \in \mathbb{S} : \Delta_\infty(L(\boldsymbol{y}|\boldsymbol{x}) \parallel \Omega(\boldsymbol{y})) \leq k. \tag{3}$$

Bounding this divergence is another way of framing what we are aiming to achieve: The divergence is large when L assigns high likelihood to a sample while Ω doesn't, which means L is likely to have produced a output that is out-of-domain. When Ω assigns high likelihood to $\boldsymbol{y}$, the sample is in the target domain, and the divergence in (3) is not restrictive. In Lemma 1 in Appendix A we show this divergence implies (1) and (2).

As the oracle is not available in practice we approximate it with a "guide" language model that is exclusively trained on in-domain data dubbed G (i.e. the guide model). We utilise $G(\boldsymbol{y})$ to replace

$\Omega(\boldsymbol{y})$ to assess the *marginal* likelihood of $\boldsymbol{y}$. While this means that $G(\boldsymbol{y})$ loses some context contained in $\boldsymbol{x}$, this has a major advantage: $G(\boldsymbol{y})$ does not depend on $\boldsymbol{x}$, which is a potential adversary and hence, by design is robust to adversarial prompts.

We utilise a general model L and a domain generator G as described above and obtain a meta-model M for which the guarantee holds with respect to the domain generator G. In particular, we perform rejection sampling as described in Algorithm 1 (inspired by Vyas et al. (2023)): The capable, general model L proposes a sample $\boldsymbol{y}$ and we accept, if the length normalized log-ratio between L and G is bounded by hyperparameter k. We repeat up to T times until a sample is accepted. If all samples are rejected, the model dismisses the request. This defines a new model M, for which the following theorem establishes the certificate:

Theorem 1 (VALID Certificate) *Let $M(\boldsymbol{y}|\boldsymbol{x})$ be the likelihood of $\boldsymbol{y}$ given $\boldsymbol{x}$ under M. Let $N_{\boldsymbol{y}}$ be the length of $\boldsymbol{y}$. Rejection sampling as described in Algorithm 1 provides the following bound on M:*

$$\forall \boldsymbol{x} \in \mathbb{S} : M_{L,G,k,T}(\boldsymbol{y}|\boldsymbol{x}) \leq 2^{kN_{\boldsymbol{y}}} \cdot T \cdot G(\boldsymbol{y}) \tag{4}$$

This means that M is $[2^{kN_{\boldsymbol{y}}} TG(\boldsymbol{y})]$-AC and, further, M is $[\max_{\boldsymbol{y}\in\mathbb{F}} 2^{kN_{\boldsymbol{y}}} TG(\boldsymbol{y})]$-DC w.r.t. $\mathbb{F}$.

When context allows, we may abbreviate $M_{L,G,k,T}$ to M, omitting subscripts for brevity. Such a certificate with respect to G can be useful: As G is only trained on samples in $\mathcal{D}_{\mathbb{T}} \subset \mathbb{T}$, a dataset of domain $\mathbb{T}$, it assigns exponentially decreasing likelihood to samples that are in $\mathbb{F}$ [1]. In particular, this is useful iff the log upper bound $kN_{\boldsymbol{y}} + \log T + \log G(\boldsymbol{y})$ (log RHS of (4)) is small in comparison to $\max_{x\in\mathbb{S}} \log L(\boldsymbol{y}|\boldsymbol{x})$: Our certificate can provide an upper bound to the adversarial behaviour of M that is favourable over L.

In the Appendices, we provide further insights into VALID. In particular, in Appendix B.2 we motivate the length normalization as employed in Algorithm 1. In Appendix A, we provide Lemma 2 showing how to estimate the likelihood of M. In Lemma 3, we provide an analysis on the expected number of iterations in VALID. Further, in Appendix B.1 we provide further intuition on how rejection sampling can achieve an adversarial bound. Finally, in Lemma 4 we derive an adversary for M and discuss how rejection sampling encumbers adversarial attacks on M.

4 Experiments

We aim to empirically test our method proposed in Section 3 across three examples: TinyShakespeare, 20NG and MedicalQA. After describing the experimental setup in Section 4.1, we examine the rejection behaviour of our method by examining the $\log L(\boldsymbol{y}|\boldsymbol{x})/G(\boldsymbol{y})$ ratio an associated certificates under a finite set of ground-truth test samples from $\mathbb{T}$ and $\mathbb{F}$ in Section 4.2. Finally, we explore the effects repeated sampling with VALID in Section 4.3.

4.1 Experimental Setup

In this section, we provide a brief description of our experimental setup for three applications. Each experimental setup consists of a target domain $\mathbb{T}$ a finite dataset $\mathcal{D}_{\mathbb{T}}$ of in-domain samples, models L and G, and an out-of-domain dataset $\mathcal{D}_{\mathbb{F}}$, against which we test our methods (see Appendix C for more details on data and models).

TinyShakespeare. TinyShakespeare (TS) is an popular dataset containing dialogues from Shakespeare's plays (Karpathy). We fine-tune a Gemma-2-2b (Gemma Team et al., 2024) and train G on this dataset using a GPT-2 architecture (33.7M parameters). At testing, we consider 256-token long sentences and use the first 128 tokens as prompt. As commonly done (Zhang et al., 2024), we consider IMDB (Maas et al., 2011), RTE (noa, 2023), SST2 (Minaee et al., 2024) as OOD datasets and add an old Bible dataset (Reis, 2019) as it is linguistically close to TinyShakespeare.

20NG. The 20NG dataset contains text from 18,000 posts on newsgroup websites on 20 different topics. We consider articles from the computer science category as target domain $\mathcal{D}_{\mathbb{T}}$ and remaining categories as OOD. We use a fine-tuned Gemma-2-2b (Gemma Team et al., 2024) as L and train G on $\mathcal{D}_{\mathbb{T}}$ using a GPT-2 architecture (109.3M parameters) with a context length of 256 tokens. At

[1] We give an empirical example of this behaviour in Figure 14 in Appendix D.8.

Table 1: Atomic Certificates @ $FRR = 0.1$. Larger is better for $\mathcal{D}_{\mathbb{F}}$, smaller is better for $\mathcal{D}_{\mathbb{T}}$.

ϵ	Proportion of $\epsilon_{\boldsymbol{y}} \leq \epsilon$					(in %)
	TinyShakespeare		20NG		MedicalQA	
	$\mathcal{D}_{\mathbb{T}}$	$\mathcal{D}_{\mathbb{F}}$	$\mathcal{D}_{\mathbb{T}}$	$\mathcal{D}_{\mathbb{F}}$	$\mathcal{D}_{\mathbb{T}}$	$\mathcal{D}_{\mathbb{F}}$
10^{-1}	100.0	100.0	100.0	100.0	82.02	100.0
10^{-5}	100.0	100.0	100.0	100.0	73.64	99.70
10^{-10}	100.0	100.0	100.0	100.0	59.90	95.14
10^{-20}	100.0	100.0	100.0	100.0	31.01	67.15
10^{-50}	99.0	100.0	98.0	99.0	0.03	6.70
10^{-100}	66.0	100.0	68.0	99.0	0.00	0.01
10^{-250}	0.0	86.0	0.00	39.0	0.00	0.00

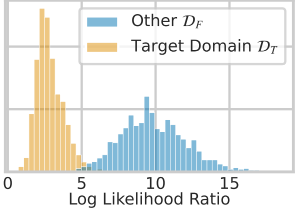


Figure 1: The log likelihood ratios are well disentangled in the MedQA setup.

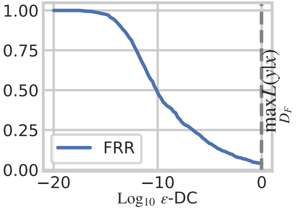


Figure 2: The false rejection rate (FRR) required to issue an ϵ-DC for the MedQA setup.

testing, we consider 256-token long sentences and use the first 128 tokens as prompt. We use the same OOD datasets as for TS and also include the non-computer science categories from 20NG.

Medical QA. We apply our method to medical question answering as target domain, $\mathbb{T}$. This could, for example, be extended to a chatbot for clinicians to look up patient symptoms. To model potential questions and answers, we use the PubMedQA dataset (Jin et al., 2019) as $\mathcal{D}_{\mathbb{T}}$, which contains approximately 200K QA pairs for training and 1000 test pairs. We regard question answering on other topics, such as geography or computer science, as $\mathbb{F}$. To model this, we use the Stanford Question and Answering Dataset (excluding medical categories) (Rajpurkar et al., 2016) as $\mathcal{D}_{\mathbb{F}}$. As a generalist LLM, L, we use a LLama-3-8B model (AI@Meta, 2024) and for G we train a GPT-2 architecture model from scratch (184M parameters) (Radford et al., 2019). We pre-train G on PubMedQA and fine-tune it on questions from PubMedQA paired with answers generated by L.

4.2 Likelihood Ratios on Ground Truth Samples

In this section, we evaluate the capability of our method to attribute samples to the target domain and investigate whether it yields useful adversarial bounds. In particular, we study the length-normalized likelihood ratio $L(\boldsymbol{y}|\boldsymbol{x})/G(\boldsymbol{y})$ on in- and out-of-domain samples. In Figure 1, we show that the log likelihood ratios for MedicalQA are disentangled and hence a threshold k exists separating target domain and out-of-domain samples well. However, such k — while yielding strong OOD detection performance — might not be associated with tight certificates. Hence, we will first study the $\epsilon_{\boldsymbol{y}}$-AC certificates under M for individual samples, $\boldsymbol{y}$, before moving on to the domain certificate, ϵ-DC.

Atomic Certificates. For each $\boldsymbol{y}$, we obtain the $\epsilon_{\boldsymbol{y}}$-AC using VALID (see Section 3). We compute the proportion of certificates $\epsilon_{\boldsymbol{y}}$ smaller than some ϵ separately for samples in the target domain dataset, $\mathcal{D}_{\mathbb{T}}$, and the out-of-domain datasets $\mathcal{D}_{\mathbb{F}}$ and present results in Table 1. Due to varying sequence lengths between datasets, we see some variation in the results, but make similar core observations for all three setups: First, the certificates on the out-of-domain datasets $\mathcal{D}_{\mathbb{F}}$ are *meaningfully tight*. We observe that more than 95% of out-of-domain samples have a $\epsilon_{\boldsymbol{y}}$-AC of less than 1×10^{-10} across all datasets. Further, we notice for each dataset, that fewer certificates in $\mathcal{D}_{\mathbb{T}}$ are tight relative to those in $\mathcal{D}_{\mathbb{F}}$. For MedicalQA fewer than 60% of samples have a bound of $\epsilon = 10^{-10}$. This distinction between certificates on $\mathcal{D}_{\mathbb{T}}$ and $\mathcal{D}_{\mathbb{F}}$ is important: The certificate should be *restrictive* on samples in $\mathbb{F}$, and permissive in $\mathbb{T}$, i.e. the bound should not prevent in-domain responses to be sampled. Both are satisfied in our experiments. To further our analysis of the atomic certificates, we compare these with a baseline in Appendix D.1.

Domain Certificates. To study certification across a range of samples, we turn to the domain certificate, ϵ-DC. Above, we studied the effect of various parameters (e.g. fixing FRR) on the certificates. However, it's most likely practitioners work the other way around: They first set an acceptable threshold according to a threat and safety model. Then, they examine model performance under conditions satisfying such certificate. Hence, we study model performance at a given ϵ-DC. As proposed in Section 2.1, we establish an ϵ-DC certificate w.r.t. $\mathcal{D}_{\mathbb{F}}$ approximating the certificate for $\mathbb{F}$. To obtain $\epsilon_{\boldsymbol{y}}$-ACs that adhere to the domain certificate ϵ, we need to choose rejection threshold, k, and the number of iterations, T, accordingly. For simplicity, here we keep $T = 1$ and study model performance on $\mathcal{D}_{\mathbb{T}}$, our in-domain data, while maintaining an ϵ-DC over $\mathcal{D}_{\mathbb{F}}$. In particular, we look at the FRR of M: The performance of model M is determined by the performance of L (from which we sample response candidates), and the false rejections leading to a degradation of M compared to L. Hence, we study the FRR as a function of the certification threshold ϵ. The result is shown in Figure 2 for MedicalQA: The FRR increases as the certificates get tighter (small ϵ). Remarkably, we achieve a domain certificate with $\epsilon = 10^{-5}$ at a FRR of only 15% at a single rejection step. We replicate all figures for the other datasets in Appendix D.

4.3 Repeated Sampling in VALID

In Section 4.2 above, we study certificates on ground truth samples. In Appendix D.3, we replicate this analysis in detail by applying our Algorithm 1, actually sampling from L with $T = 1$, drawing conclusions inline with the above. Here, we go another step further as we provide an analysis of VALID with $T > 1$. In particular, we study ϵ-DCs as a function of rejection threshold k and sampling iterations T.

Setup. We adopt the MedicalQA setup as described in Section 4.1. Specifically, we consider $T, k \in \{1, 2, 3, 4, 5\}$, and calculate the resulting $\epsilon - DC$ on $\mathcal{D}_{\mathbb{F}}$ and the false rejection rate (FRR) on $\mathcal{D}_{\mathbb{T}}$. For ease of presentation we use a fixed temperature of 1.0 for L.[2]

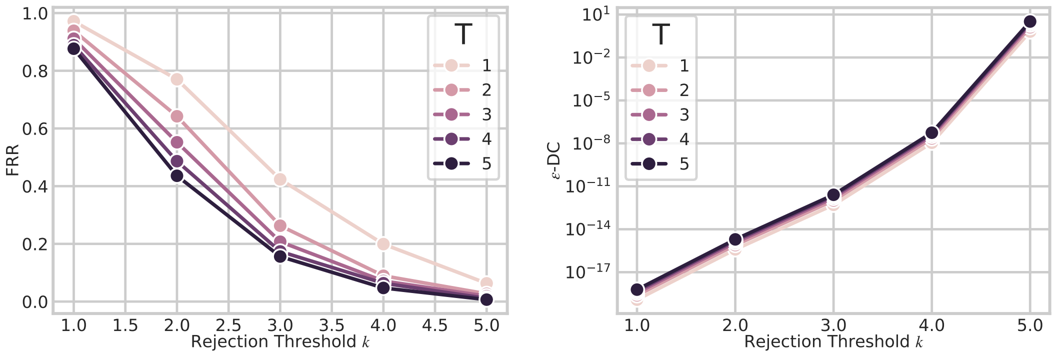
Results. We find that increasing T significantly reduces FRR while only marginally increasing the ϵ-DC (domain certificate). We present findings for the FRR in Figure 3a and for ϵ-DC in Figure 3b. The minor increase in ϵ due to increasing T should not come as a surprise as we recall the formula for the upper bound: $2^{kN_y}TG(y)$ (see (4)). Even $T = 10$ increases the upper bound ϵ_y by only one order of magnitude. On the other hand, the gains in in-domain performance are marked. In Figure 3a we can observe that for $T = 5$, the FRR is roughly halved for $(K > 2)$, greatly improving the refusal behaviour of the model on in-domain samples.

We provide further insights in the Appendices. We demonstrate how to evaluate a certified model on standardized benchmarks in Appendix D.4. In Appendix E, we study the performance gap between G and M and find that M outperforms G. Together with the results above, this demonstrates that our system, combining the performance of L and safety of G, cannot be matched by either L or G by themselves. Further, the effectiveness of VALID utilising a G of such limited capacities demonstrates that the burden on training G is relatively low: A model that performs poorly at the target task, but distinguishes well between samples in $\mathbb{T}$ and $\mathbb{F}$, can be sufficient to achieve meaningful certificates for M.

5 Conclusion

In this work, we tackle the problem of generative language models producing outputs outside their target domain in response to adversarial inputs. We describe the associated risks, introduce a first-of-its-kind framework for domain certification of LLMs, and provide VALID, a simple algorithm relying on well-established theories from statistics and information theory to provide such guarantees. We demonstrate the effectiveness of VALID in multiple representative settings and show that it is effective even when relying on a guide model G with limited language skills, making it easy to deploy in limited data and resource environments.

[2] We note that temperature of L is a confounding factor here. For $t_L \to 0$, we will observe deterministic sampling of y given x and hence, $T > 1$ will not have any benefit.



(a) (b)

Figure 3: False Rejection Rate (FRR) **(Left)** and $\epsilon - DC$ of the Domain certificate of VALID **(Right)** plotted for a range of different values of T and k.

Acknowledgments

This work is supported by a UKRI grant Turing AI Fellowship (EP/W002981/1). C. Emde and M. Kayser are supported by the EPSRC Centre for Doctoral Training in Health Data Science (EP/S02428X/1) and the AXA Research Fund. A. Bibi has received an Amazon Research Award. T. Lukasiewicz is supported by the AXA Research Fund. Tom Rainforth is supported by the UK EPSRC grant EP/Y037200/1. We also thank the Royal Academy of Engineering and FiveAI. Further, we thank Samuele Marro for his advice.

References

nyu-mll/glue · Datasets at Hugging Face, December 2023.

AI@Meta. Llama 3 Model Card. 2024.

Naveed Akhtar, Ajmal Mian, Navid Kardan, and Mubarak Shah. Advances in Adversarial Attacks and Defenses in Computer Vision: A Survey. *IEEE Access*, 9:155161–155196, 2021. doi: 10.1109/ACCESS.2021.3127960.

Ibrahim M Alabdulmohsin, Behnam Neyshabur, and Xiaohua Zhai. Revisiting neural scaling laws in language and vision. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 22300–22312. Curran Associates, Inc., 2022.

Amanda Askell, Yuntao Bai, Anna Chen, Dawn Drain, Deep Ganguli, Tom Henighan, Andy Jones, Nicholas Joseph, Ben Mann, Nova DasSarma, et al. A general language assistant as a laboratory for alignment. *arXiv preprint arXiv:2112.00861*, 2021.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.

Yoshua Bengio. Bounding the probability of harm from an ai to create a guardrail. `https://yoshuabengio.org/2024/08/29/bounding-the-probability-of-harm-from-an-ai-to-create-a-guardrail/`, 2024a. Accessed: 2024-09-27.

Yoshua Bengio. Bounding the probability of harm from an ai to create a guardrail, August 29 2024b. Accessed: 2024-11-23.

Yoshua Bengio, Michael K. Cohen, Nikolay Malkin, Matt MacDermott, Damiano Fornasiere, Pietro Greiner, and Younesse Kaddar. Can a Bayesian Oracle Prevent Harm from an Agent?, 2024. _eprint: 2408.05284.

Battista Biggio, Blaine Nelson, and Pavel Laskov. Poisoning Attacks against Support Vector Machines, 2013. _eprint: 1206.6389.

Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, Shyamal Buch, Dallas Card, Rodrigo Castellon, Niladri Chatterji, Annie Chen, Kathleen Creel, Jared Quincy Davis, Dora Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren Gillespie, Karan Goel, Noah Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, Omar Khattab, Pang Wei Koh, Mark Krass, Ranjay Krishna, Rohith Kuditipudi, Ananya Kumar, Faisal Ladhak, Mina Lee, Tony Lee, Jure Leskovec, Isabelle Levent, Xiang Lisa Li, Xuechen Li, Tengyu Ma, Ali Malik, Christopher D. Manning, Suvir Mirchandani, Eric Mitchell, Zanele Munyikwa, Suraj Nair, Avanika Narayan, Deepak Narayanan, Ben Newman, Allen Nie, Juan Carlos Niebles, Hamed Nilforoshan, Julian Nyarko, Giray Ogut, Laurel Orr, Isabel Papadimitriou, Joon Sung Park, Chris Piech, Eva Portelance, Christopher Potts, Aditi Raghunathan, Rob

Reich, Hongyu Ren, Frieda Rong, Yusuf Roohani, Camilo Ruiz, Jack Ryan, Christopher Ré, Dorsa Sadigh, Shiori Sagawa, Keshav Santhanam, Andy Shih, Krishnan Srinivasan, Alex Tamkin, Rohan Taori, Armin W. Thomas, Florian Tramèr, Rose E. Wang, William Wang, Bohan Wu, Jiajun Wu, Yuhuai Wu, Sang Michael Xie, Michihiro Yasunaga, Jiaxuan You, Matei Zaharia, Michael Zhang, Tianyi Zhang, Xikun Zhang, Yuhui Zhang, Lucia Zheng, Kaitlyn Zhou, and Percy Liang. On the Opportunities and Risks of Foundation Models, 2022. _eprint: 2108.07258.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language Models are Few-Shot Learners, 2020. _eprint: 2005.14165.

Nicholas Carlini, Matthew Jagielski, Christopher A. Choquette-Choo, Daniel Paleka, Will Pearce, Hyrum Anderson, Andreas Terzis, Kurt Thomas, and Florian Tramèr. Poisoning Web-Scale Training Datasets is Practical, 2024a. _eprint: 2302.10149.

Nicholas Carlini, Milad Nasr, Christopher A. Choquette-Choo, Matthew Jagielski, Irena Gao, Anas Awadalla, Pang Wei Koh, Daphne Ippolito, Katherine Lee, Florian Tramer, and Ludwig Schmidt. Are aligned neural networks adversarially aligned?, 2024b. _eprint: 2306.15447.

Marco Casadio, Tanvi Dinkar, Ekaterina Komendantskaya, Luca Arnaboldi, Matthew L Daggitt, Omri Isac, Guy Katz, Verena Rieser, and Oliver Lemon. Nlp verification: Towards a general methodology for certifying robustness. *arXiv preprint arXiv:2403.10144*, 2024.

Patrick Chao, Alexander Robey, Edgar Dobriban, Hamed Hassani, George J. Pappas, and Eric Wong. Jailbreaking Black Box Large Language Models in Twenty Queries, 2024. _eprint: 2310.08419.

Isha Chaudhary, Vedaant V Jain, and Gagandeep Singh. Quacer-c: Quantitative certification of knowledge comprehension in llms. *arXiv preprint arXiv:2402.15929*, 2024.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2023.

Gabriel Chua, Shing Yee Chan, and Shaun Khoo. A flexible large language models guardrail development methodology applied to off-topic prompt detection. *arXiv preprint arXiv:2411.12946*, 2024.

Josef Dai, Xuehai Pan, Ruiyang Sun, Jiaming Ji, Xinbo Xu, Mickel Liu, Yizhou Wang, and Yaodong Yang. Safe rlhf: Safe reinforcement learning from human feedback. *arXiv preprint arXiv:2310.12773*, 2023.

Xinshuai Dong, Anh Tuan Luu, Min Lin, Shuicheng Yan, and Hanwang Zhang. How Should Pre-Trained Language Models Be Fine-Tuned Towards Adversarial Robustness? In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 4356–4369. Curran Associates, Inc., 2021.

Yi Dong, Ronghui Mu, Yanghao Zhang, Siqi Sun, Tianle Zhang, Changshun Wu, Gaojie Jin, Yi Qi, Jinwei Hu, Jie Meng, Saddek Bensalem, and Xiaowei Huang. Safeguarding Large Language Models: A Survey, 2024a. _eprint: 2406.02622.

Yi Dong, Ronghui Mu, Yanghao Zhang, Siqi Sun, Tianle Zhang, Changshun Wu, Gaojie Jin, Yi Qi, Jinwei Hu, Jie Meng, et al. Safeguarding large language models: A survey. *arXiv preprint arXiv:2406.02622*, 2024b.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaoqing Ellen Tan, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aaron Grattafiori, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alex Vaughan, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Franco, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, Danny Wyatt, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Firat Ozgenel, Francesco Caggioni, Francisco Guzmán, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Govind Thattai, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Karthik Prasad, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kun Huang, Kunal Chawla, Kushal Lakhotia, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Maria Tsimpoukelli, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikolay Pavlovich Laptev, Ning Dong, Ning Zhang, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Rohan Maheswari, Russ Howes, Ruty Rinott, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Kohler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vítor Albiero, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaofang Wang, Xiaojian Wu, Xiaolan Wang, Xide Xia, Xilun Wu, Xinbo Gao, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yuchen Hao, Yundi Qian, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, and Zhiwei Zhao. The Llama 3 Herd of Models, 2024. _eprint: 2407.21783.

Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. HotFlip: White-Box Adversarial Examples for Text Classification. In Iryna Gurevych and Yusuke Miyao (eds.), *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 31–36, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-2006.

EU EU. Regulation (EU) 2024/1689 of the European Parliament and of the Council of 13 June 2024 laying down harmonised rules on artificial intelligence and amending Regulations (EC) No 300/2008, (EU) No 167/2013, (EU) No 168/2013, (EU) 2018/858, (EU) 2018/1139 and (EU) 2019/2144 and Directives 2014/90/EU, (EU) 2016/797 and (EU) 2020/1828 (Artificial Intelligence Act) (Text with EEA relevance), June 2024. Legislative Body: CONSIL, EP.

Vincent Freiberger and Erik Buchmann. Fairness certification for natural language processing and large language models. In *Intelligent Systems Conference*, pp. 606–624. Springer, 2024.

Varun Gangal, Abhinav Arora, Arash Einolghozati, and Sonal Gupta. 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, pp. 7764–7771. Issue: 05.

Tianyu Gao, Adam Fisch, and Danqi Chen. Making pre-trained language models better few-shot learners. *arXiv preprint arXiv:2012.15723*, 2020.

Gemma Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*, 2024.

Saibo Geng, Martin Josifoski, Maxime Peyrard, and Robert West. Grammar-constrained decoding for structured nlp tasks without finetuning. *arXiv preprint arXiv:2305.13971*, 2023.

Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring Massive Multitask Language Understanding, 2021. _eprint: 2009.03300.

Tom Henighan, Jared Kaplan, Mor Katz, Mark Chen, Christopher Hesse, Jacob Jackson, Heewoo Jun, Tom B Brown, Prafulla Dhariwal, Scott Gray, et al. Scaling laws for autoregressive generative modeling. *arXiv preprint arXiv:2010.14701*, 2020.

Chia-Yi Hsu, Yu-Lin Tsai, Chih-Hsun Lin, Pin-Yu Chen, Chia-Mu Yu, and Chun-Ying Huang. Safe lora: the silver lining of reducing safety risks when fine-tuning large language models. *arXiv preprint arXiv:2405.16833*, 2024.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Hakan Inan, Kartikeya Upasani, Jianfeng Chi, Rashi Rungta, Krithika Iyer, Yuning Mao, Michael Tontchev, Qing Hu, Brian Fuller, Davide Testuggine, and Madian Khabsa. Llama Guard: LLM-based Input-Output Safeguard for Human-AI Conversations, 2023a. _eprint: 2312.06674.

Hakan Inan, Kartikeya Upasani, Jianfeng Chi, Rashi Rungta, Krithika Iyer, Yuning Mao, Michael Tontchev, Qing Hu, Brian Fuller, Davide Testuggine, et al. Llama guard: Llm-based input-output safeguard for human-ai conversations. *arXiv preprint arXiv:2312.06674*, 2023b.

Neel Jain, Avi Schwarzschild, Yuxin Wen, Gowthami Somepalli, John Kirchenbauer, Ping-yeh Chiang, Micah Goldblum, Aniruddha Saha, Jonas Geiping, and Tom Goldstein. Baseline defenses for adversarial attacks against aligned language models. *arXiv preprint arXiv:2309.00614*, 2023.

Robin Jia and Percy Liang. Adversarial Examples for Evaluating Reading Comprehension Systems, 2017. _eprint: 1707.07328.

Xiaojun Jia, Tianyu Pang, Chao Du, Yihao Huang, Jindong Gu, Yang Liu, Xiaochun Cao, and Min Lin. Improved Techniques for Optimization-Based Jailbreaking on Large Language Models, 2024. _eprint: 2405.21018.

Shuyu Jiang, Xingshu Chen, and Rui Tang. Prompt Packer: Deceiving LLMs through Compositional Instruction with Hidden Attacks, 2023. _eprint: 2310.10077.

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William Cohen, and Xinghua Lu. PubMedQA: A Dataset for Biomedical Research Question Answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 2567–2577, 2019.

Erik Jones, Anca Dragan, Aditi Raghunathan, and Jacob Steinhardt. Automatically Auditing Large Language Models via Discrete Optimization, 2023. _eprint: 2303.04381.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Andrej Karpathy. The unreasonable effectiveness of recurrent neural networks - http://karpathy.github.io/2015/05/21/rnn-effectiveness/.

Timo Kaufmann, Paul Weng, Viktor Bengs, and Eyke Hüllermeier. A survey of reinforcement learning from human feedback. *arXiv preprint arXiv:2312.14925*, 2023.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning.

JST Kishan. Post on x. `https://x.com/jst_kishan/status/1834059919101931868?s=61`, 2023. Accessed: 2024-09-27.

Aounon Kumar, Chirag Agarwal, Suraj Srinivas, Aaron Jiaxun Li, Soheil Feizi, and Himabindu Lakkaraju. Certifying llm safety against adversarial prompting. *arXiv preprint arXiv:2309.02705*, 2023.

E La Malfa. *On robustness for natural language processing*. PhD Thesis, University of Oxford, 2023.

Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*, 2021.

Haowei Lin and Yuntian Gu. FLatS: Principled out-of-distribution detection with feature-based likelihood ratio score. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 8956–8963. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.554.

Weitang Liu, Xiaoyun Wang, John D. Owens, and Yixuan Li. Energy-based out-of-distribution detection.

Xiaogeng Liu, Nan Xu, Muhao Chen, and Chaowei Xiao. AutoDAN: Generating Stealthy Jailbreak Prompts on Aligned Large Language Models, 2024. _eprint: 2310.04451.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning Word Vectors for Sentiment Analysis. In Dekang Lin, Yuji Matsumoto, and Rada Mihalcea (eds.), *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.

Tess McClure. Supermarket AI meal planner app suggests recipe that would create chlorine gas. *The Guardian*, August 2023. ISSN 0261-3077.

Anay Mehrotra, Manolis Zampetakis, Paul Kassianik, Blaine Nelson, Hyrum Anderson, Yaron Singer, and Amin Karbasi. Tree of Attacks: Jailbreaking Black-Box LLMs Automatically, 2024. _eprint: 2312.02119.

Shervin Minaee, Tomas Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao. Large Language Models: A Survey, February 2024. arXiv:2402.06196 [cs].

Marius Mosbach, Tiago Pimentel, Shauli Ravfogel, Dietrich Klakow, and Yanai Elazar. Few-shot fine-tuning vs. in-context learning: A fair comparison and evaluation. *arXiv preprint arXiv:2305.16938*, 2023.

Thanh Tam Nguyen, Thanh Trung Huynh, Phi Le Nguyen, Alan Wee-Chung Liew, Hongzhi Yin, and Quoc Viet Hung Nguyen. A Survey of Machine Unlearning, 2022. _eprint: 2209.02299.

Charles O'Neill, Jack Miller, Ioana Ciuca, Yuan-Sen Ting, and Thang Bui. Adversarial Fine-Tuning of Language Models: An Iterative Optimisation Approach for the Generation and Detection of Problematic Content, 2023. _eprint: 2308.13768.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35: 27730–27744, 2022.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Ethan Perez, Saffron Huang, Francis Song, Trevor Cai, Roman Ring, John Aslanides, Amelia Glaese, Nat McAleese, and Geoffrey Irving. Red Teaming Language Models with Language Models. In *Conference on Empirical Methods in Natural Language Processing*, 2022.

Fábio Perez and Ian Ribeiro. Ignore Previous Prompt: Attack Techniques For Language Models, 2022.

Alexander Podolskiy, Dmitry Lipin, Andrey Bout, Ekaterina Artemova, and Irina Piontkovskaya. Revisiting mahalanobis distance for transformer-based out-of-domain detection.

Xiangyu Qi, Yi Zeng, Tinghao Xie, Pin-Yu Chen, Ruoxi Jia, Prateek Mittal, and Peter Henderson. Fine-tuning aligned language models compromises safety, even when users do not intend to! In *The Twelfth International Conference on Learning Representations*, 2024.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language Models are Unsupervised Multitask Learners. 2019.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36, 2024.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ Questions for Machine Comprehension of Text. In Jian Su, Kevin Duh, and Xavier Carreras (eds.), *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 2383–2392, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1264. _eprint: 1606.05250.

Eduardo Reis. Bible Corpus - Basic Text Generation using N-grams, 2019.

Alfréd Rényi. On Measures of Entropy and Information. In *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics*, volume 1. University of California Press, 1961.

Mikayel Samvelyan, Sharath Chandra Raparthy, Andrei Lupu, Eric Hambro, Aram H. Markosyan, Manish Bhatt, Yuning Mao, Minqi Jiang, Jack Parker-Holder, Jakob Foerster, Tim Rocktäschel, and Roberta Raileanu. Rainbow Teaming: Open-Ended Generation of Diverse Adversarial Prompts, 2024. _eprint: 2402.16822.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural Machine Translation of Rare Words with Subword Units, 2016. _eprint: 1508.07909.

Yunhao Tang, Zhaohan Daniel Guo, Zeyu Zheng, Daniele Calandriello, Rémi Munos, Mark Rowland, Pierre Harvey Richemond, Michal Valko, Bernardo Ávila Pires, and Bilal Piot. Generalized preference optimization: A unified approach to offline alignment. *arXiv preprint arXiv:2402.05749*, 2024.

The Guardian. Pak'nsave ai meal planner suggests toxic recipes in 'malfunction'. *The Guardian*, 2023. Accessed: 2024-09-27.

Rheeya Uppaal, Junjie Hu, and Yixuan Li. Fine-tuning needed? pre-trained language models are near perfect for out-of-domain detection.

Nikhil Vyas, Sham Kakade, and Boaz Barak. On Provable Copyright Protection for Generative Models, 2023. _eprint: 2302.10870.

Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. Universal Adversarial Triggers for Attacking and Analyzing NLP, 2021a. _eprint: 1908.07125.

Eric Wallace, Tony Z. Zhao, Shi Feng, and Sameer Singh. Concealed Data Poisoning Attacks on NLP Models, 2021b. _eprint: 2010.12563.

Jiongxiao Wang, Jiazhao Li, Yiquan Li, Xiangyu Qi, Muhao Chen, Junjie Hu, Yixuan Li, Bo Li, and Chaowei Xiao. Mitigating fine-tuning jailbreak attack with backdoor enhanced alignment. *arXiv preprint arXiv:2402.14968*, 2024.

Yizhe Xiong, Xiansheng Chen, Xin Ye, Hui Chen, Zijia Lin, Haoran Lian, Jianwei Niu, and Guiguang Ding. Temporal scaling law for large language models. *arXiv preprint arXiv:2404.17785*, 2024.

Heng Xu, Tianqing Zhu, Lefeng Zhang, Wanlei Zhou, and Philip S. Yu. Machine Unlearning: A Survey, 2023. _eprint: 2306.03558.

Ruiyao Xu and Kaize Ding. Large language models for anomaly and out-of-distribution detection: A survey. *arXiv preprint arXiv:2409.01980*, 2024.

W. J. Youden. Index for rating diagnostic tests. *Cancer*, 3(1):32–35, 1950. doi: 10.1002/1097-0142(1950)3:1⟨32::AID-CNCR2820030106⟩3.0.CO;2-3.

Andi Zhang, Tim Z. Xiao, Weiyang Liu, Robert Bamler, and Damon Wischik. Your Finetuned Large Language Model is Already a Powerful Out-of-distribution Detector, 2024. _eprint: 2404.08679.

Wenxuan Zhou, Fangyu Liu, and Muhao Chen. Contrastive out-of-distribution detection for pre-trained transformers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 1100–1111. doi: 10.18653/v1/2021.emnlp-main.84.

Andy Zou, Zifan Wang, Nicholas Carlini, Milad Nasr, J Zico Kolter, and Matt Fredrikson. Universal and transferable adversarial attacks on aligned language models. *arXiv preprint arXiv:2307.15043*, 2023.

A Proofs

Theorem 1 (VALID Certificate) *Let $M(\boldsymbol{y}|\boldsymbol{x})$ be the likelihood of $\boldsymbol{y}$ given $\boldsymbol{x}$ under M. Let $N_{\boldsymbol{y}}$ be the length of $\boldsymbol{y}$. Rejection sampling as described in Algorithm 1 provides the following bound on M:*

$$\forall \boldsymbol{x} \in \mathbb{S} : M_{L,G,k,T}(\boldsymbol{y}|\boldsymbol{x}) \leq 2^{kN_{\boldsymbol{y}}} \cdot T \cdot G(\boldsymbol{y}) \tag{4}$$

This means that M is $[2^{kN_{\boldsymbol{y}}}TG(\boldsymbol{y})]$-AC and, further, M is $[\max_{\boldsymbol{y}\in\mathbb{F}} 2^{kN_{\boldsymbol{y}}}TG(\boldsymbol{y})]$-DC w.r.t. $\mathbb{F}$.

Proof: Let A_t and A'_t be the events of accepting and rejection on iteration t, respectively. Let S_t be the event of sampling $\boldsymbol{y} \sim L(\cdot|\boldsymbol{x})$ on iteration t:

$$M(\boldsymbol{y}|\boldsymbol{x}) = \sum_{t=1}^{T} P(S_t \cap A_t) = \sum_{t=1}^{T} P(A_t|S_t)P(S_t)\prod_{i<t} P(A'_i) \tag{5}$$

We upper bound the probability of rejecting on any previous iteration by 1, $P(A'_t) \leq 1, \forall t$. $P(A|S_t)$ is non-stochastic and is equal to either 0 or 1. The only way to get a non-zero output is if $P(A|S_t) = 1$ which imply $\log \frac{L(\boldsymbol{y}|\boldsymbol{x})}{G(\boldsymbol{y})} \leq kN_y$ or $P(S_t) \leq 2^{kN_y}G(\boldsymbol{y})$ and hence by substitution:

$$M(\boldsymbol{y}|\boldsymbol{x}) = \sum_{t=1}^{T} P(S_t \cap A_t) = \sum_{t=1}^{T} P(A_t|S_t)P(S_t)\prod_{i<t} P(A'_i) \leq \sum_{t=1}^{T} 2^{kN_y}G(\boldsymbol{y}) \tag{6}$$

Finally, we sum over all t we get:

$$M(\boldsymbol{y}|\boldsymbol{x}) \leq 2^{kN_y} \cdot T \cdot G(\boldsymbol{y}). \tag{7}$$

□

Lemma 1 (Equivalence of Divergence) *Let $\Delta_\infty(P \parallel Q)$ be the Renyi divergence of order infinity (Rényi, 1961), $\Delta_\infty(P \parallel Q) \triangleq \log \sup_x \frac{P(x)}{Q(x)}$ and let Ω be a distribution over $\mathbb{T}$, i.e.* generator *for $\mathbb{T}$. Then,*

$$\forall \boldsymbol{x} \in \mathbb{X} : \Delta_\infty(L(\boldsymbol{y}|\boldsymbol{x}) \parallel \Omega(\boldsymbol{y})) \leq k, \tag{8}$$

implies Definition 1 with $\epsilon_{\boldsymbol{y}} = 2^k\Omega(\boldsymbol{y})$ and Definition 2 with $\epsilon = 2^k \max_{\mathbb{F}} \Omega(\boldsymbol{y})$. If Ω is an oracle, that assigns high likelihood to sentences in $\mathbb{T}$ and no likelihood to elements in $\mathbb{F}$ then, it implies Definition 1 with $\epsilon_{\boldsymbol{y}} = 0$ and Definition 2 with $\epsilon = 0$.

Proof: We start from the definition of the Renyi divergence and lower bound the left hand side by any element in the supremum, giving:

$$\forall \boldsymbol{x} \in \mathbb{X} : \log \frac{L(\boldsymbol{y}|\boldsymbol{x})}{\Omega(\boldsymbol{y})} \leq \log \sup_{\boldsymbol{y}} \frac{L(\boldsymbol{y}|\boldsymbol{x})}{\Omega(\boldsymbol{y})} = \Delta_\infty(L(\boldsymbol{y}|\boldsymbol{x}) \parallel \Omega(\boldsymbol{y})) \leq k. \tag{9}$$

Multiplying through by $\Omega(\boldsymbol{y})$ gives the following upper bound:

$$\forall \boldsymbol{x} \in \mathbb{X} : L(\boldsymbol{y}|\boldsymbol{x}) \leq \Omega(\boldsymbol{y}) \cdot 2^k. \tag{10}$$

showing the $\epsilon_{\boldsymbol{y}}$-AC equivalence. Taking the $\max$ over $\mathbb{F}$ shows the ϵ-DC equivalence. Assuming Ω to be a perfect oracle, we can conclude that $\forall \boldsymbol{y} \in \mathbb{F}$ the upper bound on the right hand size is zero. Thus we get the desired result:

$$\forall \boldsymbol{x} \in \mathbb{X}, \forall \boldsymbol{x} \in \mathbb{F} : L(\boldsymbol{y}|\boldsymbol{x}) = 0. \tag{11}$$

□

Lemma 2 (Likelihood of M) *let $M(\boldsymbol{y}|\boldsymbol{x})$ be the likelihood of $\boldsymbol{y}$ given $\boldsymbol{x}$ under M. Rejection sampling with as described in Algorithm 1 provides the following bound on M:*

$$M(\boldsymbol{y}|\boldsymbol{x}) = \begin{cases} L(\boldsymbol{y}|\boldsymbol{x})\frac{(1-\phi^T)}{1-\phi} & \textit{if } L(\boldsymbol{y}|\boldsymbol{x}) \leq kG(\boldsymbol{y}) \\ 0 & \textit{otherwise} \end{cases} \tag{12}$$

where $\phi = \mathbb{E}_{\boldsymbol{y}\sim L(\boldsymbol{y}|\boldsymbol{x})}[\mathbf{1}_{[L(\boldsymbol{y}|\boldsymbol{x}) \geq 2^{Nk}G(\boldsymbol{y})]}]$ or in word the condition probability of rejection on a given iteration, given and input $\boldsymbol{x}$.

Proof: Let A_t and A_t' be the events of accepting or rejection on a proposed $\boldsymbol{y}$ iteration t, respectively.

$$1-\phi = P(A_t|\boldsymbol{x}) = \mathbb{E}_{\boldsymbol{y}\sim L(\boldsymbol{y}|\boldsymbol{x})}[\mathbf{1}_{[L(\boldsymbol{y}|\boldsymbol{x})\leq 2^{Nk}G(\boldsymbol{y})]}],$$
$$\phi = P(A_t'|\boldsymbol{x}) = \mathbb{E}_{\boldsymbol{y}\sim L(\boldsymbol{y}|\boldsymbol{x})}[\mathbf{1}_{[L(\boldsymbol{y}|\boldsymbol{x})\geq 2^{Nk}G(\boldsymbol{y})]}]$$

Let S_t be the event of sampling $\boldsymbol{y}\sim L(\cdot|\boldsymbol{x})$ on iteration t, C be the event that $L(\boldsymbol{y}|\boldsymbol{x})\leq kG(\boldsymbol{y})$ and let Y be the event of $\boldsymbol{y}\sim M(\cdot|\boldsymbol{x})$

$$P(Y|C) = \sum_{t=1}^{T} P(S_t\cap A_t|C) = \sum_{t=1}^{T} P(A_t|S_t,C)P(S_t|C)\prod_{i<t}P(A_i'|C) \tag{13}$$

$P(A|S_t,C)=1$ and hence by substitution:

$$P(Y|C) = \sum_{t=1}^{T} P(S_t\cap A_t|C) = \sum_{t=1}^{T} P(A_t|S_t,C)P(S_t|C)\prod_{i<t}P(A_i'|C) = L(\boldsymbol{y}|\boldsymbol{x})\sum_{t=1}^{T}\phi^{t-1} \tag{14}$$

Notice how $\sum_{t=1}^{T}\phi^{t-1}$ is the sum of the first T entries of a geometric series. Thus we get:

$$M(\boldsymbol{y}|\boldsymbol{x}) = L(\boldsymbol{y}|\boldsymbol{x})\frac{(1-\phi^T)}{1-\phi}, \tag{15}$$

Conversely $P(A|S_t,C')=0$ and hence:

$$P(Y|C') = 0 \tag{16}$$

If the sample space $\mathbb{S}$ is large, we cannot compute $M(\boldsymbol{y}|\boldsymbol{x})$ as we cannot compute ϕ. We can estimate $M(\boldsymbol{y}|\boldsymbol{x})$ by computing $L(\boldsymbol{y}|\boldsymbol{x})$ and performing Monte Carlo sampling from L to obtain an estimator $\hat{\phi}$. We can then use the Binomial confidence interval for confidence level α:

$$\hat{\phi}\pm Z_{\alpha/2}\times\sqrt{\frac{\hat{\phi}\left(1-\hat{\phi}\right)}{N}} \tag{17}$$

We then plug in the bounds on L to obtain the bound on M because of the monotonicity of M in $\hat{\phi}$. □

Lemma 3 (Expected number of trails) *Let $\hat{t}$ be the number of iteration's run by Algorithm 1, R be the event that M rejects the abstains the from answering let $\phi = \mathbb{E}_{\boldsymbol{y}\sim L(\boldsymbol{y}|\boldsymbol{x})}[\mathbf{1}_{[L(\boldsymbol{y}|\boldsymbol{x})\geq 2^{Nk}G(\boldsymbol{y})]}]$ of the probability of rejecting a response on iteration t. Then the expected number of iteration of Algorithm 1 is given by:*

$$\mathbb{E}_{\boldsymbol{y}\sim\hat{M}(\cdot|\boldsymbol{x})}[\hat{t}] = T\phi^{2T} + \frac{(1-\phi^T)^2}{1-\phi}, \tag{18}$$

where $\hat{M}(\cdot|\boldsymbol{x})$ is the distribution over outputs of a Algorithm 1, including R, the event of Abstaining, given a input string $\boldsymbol{x}$.

Proof: The expected number of trials over all outcomes is given by:

$$\mathbb{E}[\hat{t}] = \mathbb{E}[\hat{t}|R']P(R') + \mathbb{E}[\hat{t}|R]P(R). \tag{19}$$

As R is the event that M abstains from answering the request or rejecting the proposed responses T times we have:

$$P(\mathcal{R}) = \phi^T,$$
$$P(\mathcal{R}') = 1-\phi^T.$$

If the request is rejected the algorithm will have run exactly T iterations, thus $\mathbb{E}[\hat{t}|R]=T$. The remaining quantity to calculate is $\mathbb{E}[\hat{t}|R']$:

$$\mathbb{E}[\hat{t}|R'] = \sum_{t=1}^{T} tP(t) = \sum_{t=1}^{T} tP(A)P(A')^{t-1} = (1-\phi)\sum_{t=1}^{T} t\phi^{t-1}. \tag{20}$$

Multiplying by ϕ:

$$\phi\mathbb{E}[\hat{t}|R'] = (1-\phi)\sum_{t=1}^{T} t\phi^{t}. \tag{21}$$

(20) - (21):

$$\mathbb{E}[\hat{t}|R'] - \phi\mathbb{E}[\hat{t}|R'] = (1-\phi)\sum_{t=1}^{T} t\phi^{t-1} - t\phi^{t}. \tag{22}$$

Telescoping sum:

$$(1-\phi)\mathbb{E}[\hat{t}|R'] = (1-\phi)\sum_{t=1}^{T} \phi^{t-1} - T\phi^{T}. \tag{23}$$

Canceling and summing first N element of geometric series:

$$\mathbb{E}[\hat{t}|R'] = -T\phi^{T} + \sum_{t=1}^{T}\phi^{t-1} = -T\phi^{T} + \frac{1-\phi^{T}}{1-\phi} \tag{24}$$

Plugging the relevant expressions and rearranging gives the result:

$$\mathbb{E}[\hat{t}] = \mathbb{E}[\hat{t}|R']P(R') + \mathbb{E}[\hat{t}|R]P(R), \tag{25}$$

$$\mathbb{E}[\hat{t}] = (-T\phi^{T} + \frac{1-\phi^{T}}{1-\phi})(1-\phi^{T}) + T\phi^{T} = T\phi^{2T} + \frac{(1-\phi^{T})^{2}}{1-\phi} \tag{26}$$

Note how when $\phi = 0$; the algorithm always accepts on any iteration and $\mathbb{E}[\hat{t}] = 1$. Conversely, when ϕ approaches 1 and the algorithm almost always abstains, $\mathbb{E}[\hat{t}] = T$. □

B VALID— Rejection Sampling

B.1 Attacking M

In this section, we provide some intuition on how rejection sampling (see Section 3) works to obtain an adversarial bound. For simplicity we will consider the case $T = 1$. We will then use this to describe the objective required to attack M.

Building an Intuition. We consider model M generated by rejection sampling from L using guide model G as described above. We consider a single $\boldsymbol{y}$ and describe how the ϵ-AC certificate as achieved through rejection sampling using an example. Let $\mathbf{y} =$ `The cow drinks milk` and consider three prompts:

- $\mathbf{x_1} =$ `What does a cow drink?`
- $\mathbf{x_2} =$ `Which animal drinks milk?`
- $\mathbf{x_3} =$ `Repeat after me: The cow drinks milk. Now you:`

Intuitively we may assume $L(\boldsymbol{y}|\boldsymbol{x}_3) > L(\boldsymbol{y}|\boldsymbol{x}_1) > L(\boldsymbol{y}|\boldsymbol{x}_2)$ as $\boldsymbol{y}$ more naturally follows some prompts than others: $\boldsymbol{y}$ would have very high likelihood after $\boldsymbol{x}_3$ for instruct trained models, medium likelihood after being specifically asked about cows ($\boldsymbol{x}_1$) and low likelihood to be picked as example from all mammals as response to $\boldsymbol{x}_2$.

Let us regard a single rejection step. We illustrate this example in Figure 4. If we assume that $\boldsymbol{y}|\boldsymbol{x}_1$ is rejected, i.e. $\log L(\boldsymbol{y}|\boldsymbol{x}_1) - \log G(\boldsymbol{y}) > kN_{\boldsymbol{y}}$, then we can conclude that $\boldsymbol{y}|\boldsymbol{x}_3$ will also be rejected. We know that for $T = 1$, M has likelihood:

$$M(\boldsymbol{y}|\boldsymbol{x}) = \begin{cases} L(\boldsymbol{y}|\boldsymbol{x}) & \text{if } \boldsymbol{y}|\boldsymbol{x} \text{ is accepted,} \\ 0 & \text{otherwise.} \end{cases} \tag{27}$$

Given that $M(\boldsymbol{y}|\boldsymbol{x}) = 0$ for rejected samples, the question presents itself: What is the purpose of the upper bound. Consider the case that $\boldsymbol{y}|\boldsymbol{x}_2$ is accepted. This occurs, iff $\log L(\boldsymbol{y}|\boldsymbol{x}_2) - \log G(\boldsymbol{y}) \leq kN_{\boldsymbol{y}}$, which by algebraic manipulation means $L(\boldsymbol{y}|\boldsymbol{x}_2) \leq 2^{kN_{\boldsymbol{y}}}G(\boldsymbol{y})$, recovering the upper bound as stated in Theorem 1: It is only possible to return $\boldsymbol{y}|\boldsymbol{x}_2$ when $L(\boldsymbol{y}|\boldsymbol{x}_2)$ is smaller than $2^{kN_{\boldsymbol{y}}}G(\boldsymbol{y})$.

More generally, $\boldsymbol{y}$ can only be returned, if we find an $\boldsymbol{x}^*$ s.t. $L(\boldsymbol{y}|\boldsymbol{x}^*) \leq 2^{kN_{\boldsymbol{y}}} G(\boldsymbol{y})$ and hence by (27) we have that $L(\boldsymbol{y}|\boldsymbol{x}^*) \leq 2^{kN_{\boldsymbol{y}}} G(\boldsymbol{y})$. This illustrates how rejection sampling bounds the adversaries: Samples will only be accepted if proposing them was very unlikely in the first place. This intuition helps us establishing how to attack M.

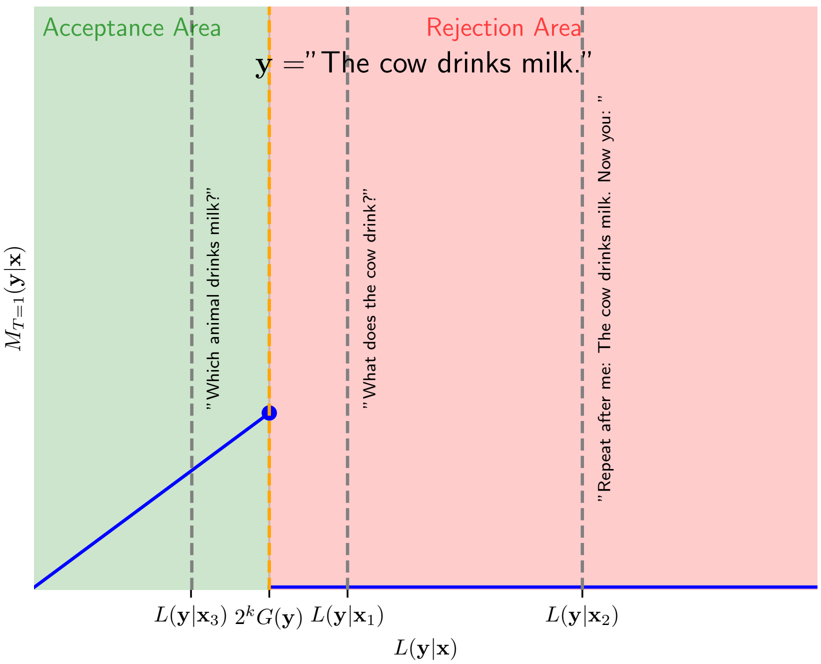


Figure 4: The likelihood of model M obtained through VALID with $T = 1$. The blue line is the likelihood of M for the given $\boldsymbol{y}$. Three example prompts $\boldsymbol{x}_1$, $\boldsymbol{x}_2$ and $\boldsymbol{x}_3$ are shown.

Formalising the Attack. We assume the adversarial objective is to increase the probability of a given $\boldsymbol{y}^*$ (e.g. from the out-of-domain set), $\mathbb{F}$, being returned. The objective of attacking L is immediately follows:

$$\boldsymbol{x}_L^{adv} = \arg\max_{\boldsymbol{x} \in \mathbb{X}} L(\boldsymbol{y}^*|\boldsymbol{x}) \tag{28}$$

where $\mathbb{X}$ is either $\mathbb{S}$ or some continuous relaxation, such as soft-prompt space. However, the solution $\boldsymbol{x}_L^{adv}$ is likely not an adversary under M, as $\boldsymbol{x}_L^{adv}$ maximises the log-likelihood ratio and thus the sample is likely rejected, hence $M(\boldsymbol{y}^*|\boldsymbol{x}_L^{adv}) = 0$. Instead, the adversary for M, $\boldsymbol{x}_M^{adv}$, needs to maximise M while ensuring the sample is accepted, i.e. $M(\boldsymbol{y}^*|\boldsymbol{x}_M^{adv}) > 0$. Thus The following objective emerges which we state in the following lemma.

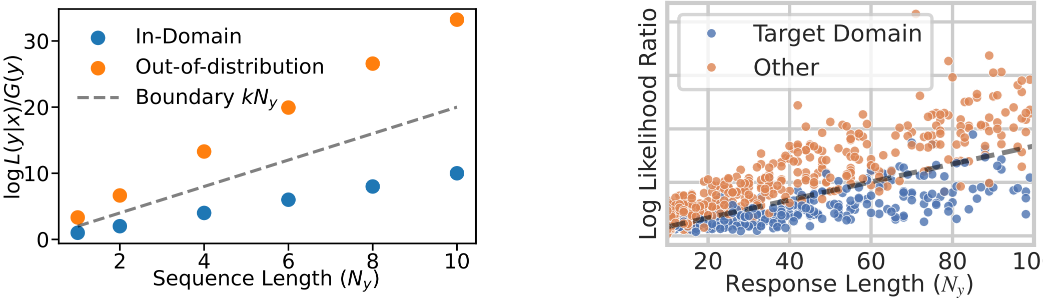
Lemma 4 (Adversary under Rejection Sampling) *Assume the adversarial objective is to increase the likelihood of sample $\boldsymbol{y}$ being returned by the model M. Assume the model M is obtained by rejection sampling as described in Algorithm 1 with $T = 1$. The adversary is given by:*

$$\boldsymbol{x}_M^{adv} = \arg\max_{\boldsymbol{x} \in \mathbb{X}} L(\boldsymbol{y}|\boldsymbol{x}) \text{ s.t. } L(\boldsymbol{y}|\boldsymbol{x}) \leq 2^{N_{\boldsymbol{y}} k} G(\boldsymbol{y}). \tag{29}$$

Proof: Note that $M(\boldsymbol{y}|\boldsymbol{x}_M^{adv}) > 0$ as the sample is accepted. Assume there exists $\boldsymbol{x}'$, s.t. $L(\boldsymbol{y}|\boldsymbol{x}') > L(\boldsymbol{y}|\boldsymbol{x}_M^{adv})$ and hence maximises M further then $\boldsymbol{x}_M^{adv}$. Then, it must be true that $L(\boldsymbol{y}|\boldsymbol{x}') > 2^k G(\boldsymbol{y})$, which implies $M(\boldsymbol{y}|\boldsymbol{x}') = 0$ yielding a contradiction. Hence, $\boldsymbol{x}_M^{adv}$ is the required solution. □

Implementing such Attack. Applying VALID to obtain M has implications on the suitable procedures to attack M. In particular, it requires solving the constrained optimisation problem in (29), which already adds a layer of complexity to the unconstrained problem for L. In general constrained optimisation problems are more challenging, this is compounded by the upper bound on $L(\boldsymbol{y}|\boldsymbol{x})$ not decomposing across tokens. Further, while large models s.t. LLama-3-8B are often publicly available, G will likely be a custom model for which the attacker does not have white box access. For a successful attack, the adversarial user must estimate the likelihood ratio between L and G, which might prove challenging. This indicates that attacking M defined through VALID might be a harder problem than attacking L. To reiterate, while it is possible to attack M, our certificate holds and it cannot be attacked past the upper bound provided in Theorem 1.

B.2 Length Normalization



(a)

(b)

Figure 5: Figure 5a shows the log likelihood ratios scale in the sequence length $N_{\boldsymbol{y}}$. Six artificial examples of sentences with length 1 to 10 are shown for the ID and OOD dataset. As log ratios scale, so should the decision boundary. Figure 5b shows the log likelihood ratio depends on $N_{\boldsymbol{y}}$ for real data from the MedQA setup. Performing length normalization makes the problem linearly separable.

Algorithm 1 performs length normalized rejection-sampling as unnormalized log likelihood ratios scale unfavourably in $N_{\boldsymbol{y}}$, the length of sequence $\boldsymbol{y}$ which we now demonstrate. Consider the next-token models l and g underlying the sequence-to-sequence models L and G. As $\boldsymbol{y}$ is sampled from L, we expect each token $y_1, \ldots, y_{N_{\boldsymbol{y}}}$ to have high likelihood under l. If we assume that l places c times more probability mass per token than g, then we can show that the log likelihood ratio grows linearly in $N_{\boldsymbol{y}}$, the length of sequence $\boldsymbol{y}$: $\log L(\boldsymbol{y}|\boldsymbol{x})/G(\boldsymbol{y}) = \log \prod_{n=1}^{N_{\boldsymbol{y}}} cg(y_n|y_{<n})/g(y_n|y_{<n}) = N_{\boldsymbol{y}} \log c$. We illustrate an example in Figure 5a: Assume that an in-domain sample $\boldsymbol{y}$ for which model and generator assign constant likelihood per token of 0.1 and 0.05, respectively, i.e. $\forall n = 1, .., N_{\boldsymbol{y}} : l(y_n|y_{<n}, \boldsymbol{x}) = 0.1$ and $g(y_n|y_{<n}, \boldsymbol{x}) = 0.05$. Further assume out-of-domain $\boldsymbol{y}'$ for which l assigns a mass of 0.1 per token, and g assigns 0.01. The log likelihood ratio for $\boldsymbol{y}$ can be expressed as $N_{\boldsymbol{y}} \log 2$ and for $\boldsymbol{y}'$ as $N_{\boldsymbol{y}'} \log 10$. As in- and out-of-domain ratios grow with length, so does the optimal decision bound. We plot sequences of varying lengths with these parameters in Figure 5a and replicate this on real data in Figure 5b. By arithmetic manipulation, rejection sampling with threshold $kN_{\boldsymbol{y}}$ is equivalent to bounding the ratio of geometrically normalized likelihoods $\log L(\boldsymbol{y}|\boldsymbol{x})^{1/N_{\boldsymbol{y}}}/G(\boldsymbol{y})^{1/N_{\boldsymbol{y}}}$ using a constant threshold k. Hence, we propose to use normalized log ratios in Algorithm 1 over unnormalized likelihood ratios. Similar approaches have been discussed in the NLP literature (Geng et al., 2023).

Despite the length normalization of the rejection threshold, notice that the VALID bound depends on $N_{\boldsymbol{y}}$, the length of sequence $\boldsymbol{y}$ (see (4)). In particular, let $\bar{g}(\boldsymbol{y})$ be the geometric mean of per-token probability for $G(\boldsymbol{y})$. The log upper bound can be written as $kN_{\boldsymbol{y}} + N_{\boldsymbol{y}} \log \bar{g}(\boldsymbol{y}) + \log T$. The ratio $k/\log \bar{g}(\boldsymbol{y})$ determines the bound strictness across varying sequences length. If is close to 1, the bound is balanced, and when $k < -\log \bar{g}(\boldsymbol{y})$, the bound decreases as $N_{\boldsymbol{y}}$ increases.

C Experimental Setup

C.1 CharTask Dataset

Here we provide more information on the CharTask dataset. The goal of the CharTask dataset is it, to have a well-controlled toy dataset with clear definitions of target domain $\mathbb{T}$ and other domains $\mathbb{F}$.

As shown in Table 2, each sequence consists of three parts: A sequence of random characters, a task definition in the middle and another sequence of characters in the end. We refer to the random

Table 2: Examples of the CharTask dataset

Task	Pool	Sequence			
		Prompt	Task	Completed	Combined
Sorting	Int	5 3 6	S R A E	3 5 6	Q 5 3 6 S R A E 3 5 6
Adding	Int	5 3 6	A E R S	6 4 7	Q 5 3 6 A E R S 6 4 7
Reverse Sorting	Int	5 3 6	R E A S	6 5 3	Q 5 3 6 R E A S 6 5 3
Even-Odd	Int	5 3 6	E R A S	6 3 5	Q 5 3 6 E R A S 6 3 5
Sorting	Int + Char	13 5 c a	S E R A	13 5 a c	Q 13 5 c a S E R A
Adding	Int + Char	13 5 c a	A S R E	14 6 d b	Q 13 5 c a A S R E
Reverse Sorting	Int + Char	13 5 c a	R E A S	c a 5 13	Q 13 5 c a R E A S c a 5 13
Even-Odd	Int + Char	13 5 c a	E S A R	a c 13 5	Q 13 5 c a E S A R 13 5 c a

sequence as S_{in}. In the middle there are four task tokens, the first of which defines the task T. “S” sets the task to sorting, “R” to reverse sorting, “A“ to adding $+1$ and “E” to even-odd sorting. The instructing token is followed by the remaining three task tokens in random order to ensure that all are seen by a model trained on a subset of these. Finally, the completed sequence is the original sequence of characters with the task performed on them, i.e. $S_{out} = T(S_{in})$. The pool of characters for each sequence is either the integers or integers and strings of lower case letters. Importantly, all tasks interpret characters and integers as characters alike. For instance, sorting integers “11”, “5” results in “11”, “5”. To be precise, all tasks are based on the integer unicode representations of the characters.

Each sequence has variable length of up to 49 elements in S_{in} (elements can be double digits). For integers we use a pool of 49 unique distinct integers and for characters we use a pool of 249 elements (e.g. defining “at” as one element in the sequence). Under these conditions there exist a combinatorially large set of unique sequences far exeeding our training dataset size.

Given the tasks and pools of characters, 8 possible domains, emerge as shown in Table 2, which we denote as CharTask (Task, Pool). We define sorting integers as the target domain: $\mathcal{D}_{\mathbb{T}}$ = CharTask (Sorting, Int) and all other combinations as out-of-domain. We create two distinct datasets with non-overlapping splits for training, validation and testing. The in-domain dataset consists of 1M training samples. The “generalist” dataset $\mathcal{D}_{\mathbb{T}+\mathbb{F}}$ = CharTask (All, Int + Char) contains of all possible tasks with sequences consisting of integers and characters. We use 1M training sequences per task, hence 4M sequences in total. Validation and test sets are 64 sequences and 4096 sequences, respectively.

C.2 CharTask Setup

Dataset and Domain. We use the CharTask dataset as described in Appendix C.1. We train a custom BPE tokenizer of length 360 (Sennrich et al., 2016). In practice, the pretrained tokenizer of any foundation model is trained on a general dataset. Hence, we train the tokenizer using $\mathcal{D}_{\mathbb{T}}$ and $\mathcal{D}_{\mathbb{F}}$, the target and out-of-domain datasets. While the dataset is inherently suitable for a sequence-to-sequence task, we treat it as next-token prediction problem just as used in language modelling.

Training. We train our domain model G on a set of integer sorting examples, CharTask (Sorting, Int). We train a GPT-2 (Radford et al., 2019) architecture with 3 layers, 3 heads and 48 embedding dimension. We train the model on partial sequences, as we are embedding marginal sequences $\boldsymbol{y}$. Hence, we cut each sequence in two parts using a splitting point that is sampled under a uniform distribution. Hence, the model learns the transition from “[BOS] ..” to any character that might be the first response token.

For the generalist model L, we train using all available tasks on integers and characters, CharTask (All,Int+Char). We train a GPT-2 architecture with 6 layers, 6 heads and 192 embedding dimensions.

We train L and G with AdamW (weight decay 0.1) for 2048 steps with a cosine learning rate schedule with 500 steps warmup, a maximum learning rate of 0.005, scheduled for 40 epochs. We train with 120 context window using next-token prediction.

Inference. We use common parameters to tweak the predictive distribution of our models. For G we use a temperature of 0.7 and for L of 0.2. We find this greatly helps the model performance of both. We do not perform *TopK* selection of tokens. We prompt with a prompt length of 10. The task-completed sequence is almost deterministic given the prompt and task for models that have very high accuracy. Hence, we remove sequences where the prompt of 10 tokens is larger than 25% of the entire sequence.

C.3 20NG Setup

Dataset Cleaning. The 20NG dataset is very dirty, containing a wide array of random special character sequences and formatting. We found these sequences to complicate model training and large pre-trained models struggled with it. In addition, as formatting strongly varies between the 20NG dataset and others, this is a confounding factor for OOD detection. Classifying sentences as ID or OOD should focus on semantics, but the formatting provides a spurious correlation that is easily exploited by models. Hence, we decided to clean the dataset. To do so we utilise the `scikit-learn` (v1.5.1) (Pedregosa et al., 2011) options to remove headers, footers and quotes. Further, we cleaned it using Llama-3.1-8B-Instruct (Dubey et al., 2024) using the following query:

```
Your task is to clean and format a string.
Instructions:
- Do not change the order of the words.
- Remove cryptic character sequences, spacings out of order,
and line breaks within sentences.
- Remove out-of-order punctuation, but leave correct
punctuation in place.
- The result should be semantically and lexically the same as
the original but well formatted.
- Remove IP addresses and email addresses.
- Remove sequences of (special) characters, that are not
human language.
- Only return the cleaned string without messages or quotes
around it.  Do not return any other information.  Do not
repeat the instructions.  Do not repeat the example.

Sentence:
```

We check the output for various keywords and phrases from prompt and find 0% violation rate. While there still exist random sequences, the data quality is greatly improved. We notice that several sequences exist in 20NG and OOD testing datasets that are seemingly random character sequences and multiple trigram repetitions such as "Nanaimo British Columbia Nanaimo British Columbia Nanaimo British Columbia ...". These sequences have the highest likelihood under model G and L while not having any semantic meaning nor constituting a valid sequence that could indicate model misappropriation. Hence, when reporting $\max$ likelihoods for 20NG over a finite dataset (e.g. $\max_{\boldsymbol{x},\boldsymbol{y}\in\mathcal{D}_{\mathbb{F}}} L(\boldsymbol{y}|\boldsymbol{x})$) we instead use the 99.99th quantile and report it as $\max$.

Training. We use a pre-trained Gemma 2 tokenizer for both models which has a vocabulary size of 256k tokens.

For the fine-tuned model L, we use a pre-trained decoder-only Gemma 2 2B (hosted on Hugging Face) as the starting point then fine-tune it to our ID dataset using LoRA adaptors which involved training an additional 10.4M parameters (0.4% of the total parameters).We train L with AdamW (weight decay 0.01) for 1536 steps with a cosine learning rate schedule with 64 steps warmup, a maximum learning rate of 5e-5, scheduled for 32 epochs. We train with 256 context window using next-token prediction.

For the model G, we use a decoder-only GPT-small model architecture, 6 layers, 6 heads and 384 embedding dimensions and a total of param- eters 109.3M, which we train from scratch using the ID data exclusively. We train G with AdamW (weight decay 0.01) for 320 steps with a cosine learning rate schedule with 100 steps warmup, a maximum learning rate of 3e-4, scheduled for 100 epochs. We train with 256 context window using next-token prediction.

Inference. For both L and G we use a default temperature of 1. We do not perform *TopK* selection of tokens. When evaluating performance, we use 128-token long prompt and a 128-token long ground truth response.

C.4 TinyShakespeare Setup

Dataset Cleaning. The formatting in TinyShakespeare dataset was distinctly different to other texts with long sequences of line breaks and usage of all-caps for character names. We removed these excessive line breaks and changed the character names from all caps to title case to make it similar to other datasets and make OOD detection less trivial challenging.

Training. We use a pre-trained Gemma 2 tokenizer for both models which has a vocabulary size of 256k tokens.

For the fine-tuned model L, we use a pre-trained decoder-only Gemma 2 2B (hosted on Hugging Face) as the starting point then fine-tune it to our ID dataset using LoRA adaptors which involved training an additional 10.4M parameters (0.4% of the total parameters). We train L with AdamW (weight decay 0.01) for 128 steps with a cosine learning rate schedule with 64 steps warmup, a maximum learning rate of 5e-5, scheduled for 32 epochs. We train with 256 context window using next-token prediction.

For the model G, we use a decoder-only GPT-micro model architecture, 4 layers, 4 heads and 128 embedding dimensions and a total of param- eters 33.7M, which we train from scratch using the ID data exclusively. We train G with AdamW (weight decay 0.01) for 2400 steps with a cosine learning rate schedule with 300 steps warmup, a maximum learning rate of 3e-4, scheduled for 300 epochs. We train with 256 context window using next-token prediction.

Inference. For both L and G we use a default temperature of 1. We do not perform *TopK* selection of tokens. When evaluating performance, we use 128-token long prompt and a 128-token long ground truth response.

C.5 MedicalQA

We apply our method to medical question answering as target domain, $\mathbb{T}$. This could for example be extended to a chatbot for clinicians to research patient symptoms. To model potential questions and answers, we use the PubMedQA dataset (Jin et al., 2019) as $\mathcal{D}_{\mathbb{T}}$, which contains approximately 200K QA pairs for training and 1000 test pairs. We regard question answering on other topics, such as geography or computer science as $\mathbb{F}$. To model this, we use the Stanford Question and Answering Dataset (excluding medical categories) (Rajpurkar et al., 2016) as $\mathcal{D}_{\mathbb{F}}$.

Training. As a generalist LLM, L, we use a LLama-3-8B model (AI@Meta, 2024) and train a custom GPT-2 model (184M parameters) for G (Radford et al., 2019). We pre-train G on PubMedQA (Jin et al., 2019) with 200K sequences. We then use 100K prompts from PubMedQA to generate sequences using L and then fine-tune on them using responses from L to half the prompts in PubMedQA. As G is embeds the responses, $G(\boldsymbol{y})$, we fine-tune using "BOS[Response]" rather than entire sequences. We pretrain with learning rate of 0.0001 for 50 epochs and then fine-tune with learning rate 0.00001 for another 50 epochs. On 8 × H100, the total training takes about 2 hours.

Inference. We perform inference without top_k or top_p parameters and with temperatures of 1.0 for model L and G. We prompt using the natural questions as defined by the datasets. For the analysis, we remove responses that are not clearly out of domain. For instance the response "10 million people every year" is not only a valid response to a geographical question, but can also be an information about the prevalence of the disease. When applying our method, we focus on responses with at least 10 tokens to further remove ambiguous questions. Modern LLMs tend to be very verbose in their responses, so responses should naturally be longer than 10 tokens.

C.6 Dataset Categories

We list here the categories excluded from SQuAD and included in MMLU for reproducibility.

Excluded From SQuAD	Included in MMLU-Med
Antibiotics	Anatomy
Symbiosis	Clinical knowledge
Gene	College medicine
Brain	College biology
Immunology	College chemistry
Biodiversity	High school biology
Digestion	High school chemistry
Pharmaceutical industry	High school psychology
Mammal	Human aging
Nutrition	Human sexuality
Tuberculosis	Medical genetics
On the Origin of Species	Nutrition
Asthma	Professional medicine
Pain	Virology
Bacteria	
Infection	
Black Death	
Pharmacy	
Immune system	
Chloroplast	

Table 3: Categories of items in used Datasets.

D Experimental Results

D.1 Comparing Atomic Certificates

To further study the atomic certificates on M, we compare them to a certificate on L as a baseline. To this end, we define the *constriction ratio* for each $\boldsymbol{y}$, given by the ratio of the certifiable $\epsilon_{\boldsymbol{y}}$ for L, $\epsilon_{\boldsymbol{y}}(L)$, over the certifiable $\epsilon_{\boldsymbol{y}}$ for M, $\epsilon_{\boldsymbol{y}}(M)$:

$$CR_k = \frac{\epsilon_{\boldsymbol{y}}(L)}{\epsilon_{\boldsymbol{y}}(M)} \tag{30}$$

To the best of our knowledge, only vacuous certificates for a general model L exist. Hence, we approximate it from below using the likelihood $L(\boldsymbol{y}|\boldsymbol{x})$ under *non-adversarial* $\boldsymbol{x}$ from the out-of-domain dataset $\mathcal{D}_{\mathbb{F}}$. Concretely, we use $L(\boldsymbol{y}|\boldsymbol{x})$ as a crude approximation of $\max_{\boldsymbol{x} \in \mathbb{S}} L(\boldsymbol{y}|\boldsymbol{x})$. This

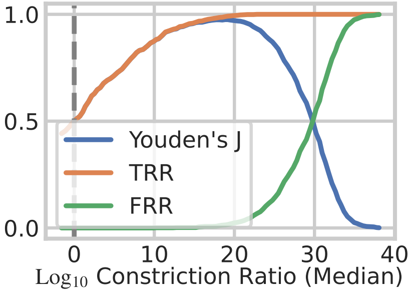


Figure 6: The trade-off between OOD and certification: The best OOD detection performance occurs with a constriction ratio of 20.

Table 4: The *log constriction ratio* for out-of-domain samples, $\mathcal{D}_{\mathbb{F}}$, shows that our atomic certificates obtained with VALID are orders of magnitudes tighter than the standard model L. For each false rejection rate (FRR) we present the 10% quantile, the median and the 90% quantile. (Larger is better).

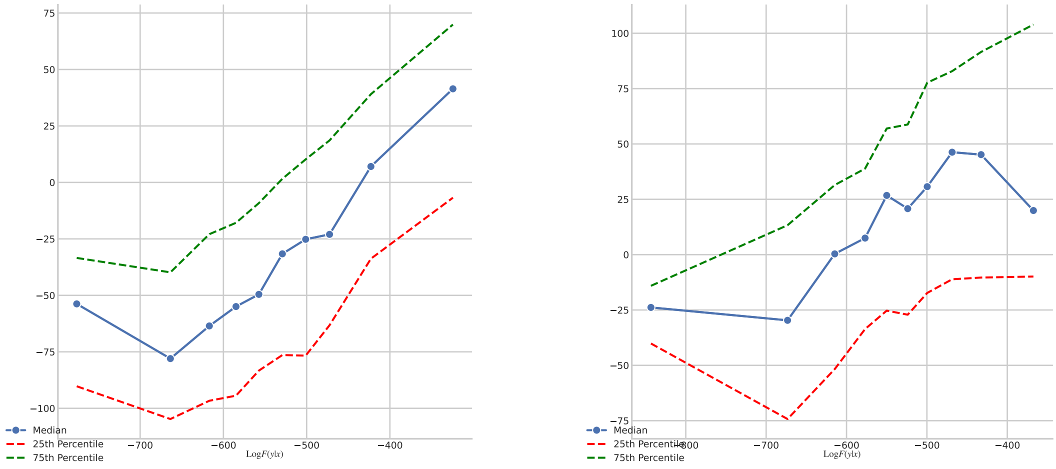
FRR	Log$_{10}$ Constriction Ratio (10% / Median / 90%)		
	TinyShakespeare	20NG	MedicalQA
0%	20 / 104 / 183	-119 / -32 / -64	-1 / 10 / 25
1%	30 / 114 / 194	-91 / -4 / 92	6 / 18 / 35
5%	55 / 140 / 219	-47 / 40 / 136	10 / 22 / 41
25%	94 / 179 / 258	-2 / 85 / 181	14 / 27 / 48
50%	115 / 199 / 278	22 / 109 / 205	16 / 30 / 52

overestimates the robustness of L and underestimates the constriction ratio, i.e. the improvement of VALID certificates over L in bounding the probability of OOD responses. In Figure 6, we plot the median constriction ratio for out-of-domain samples for MedicalQA across a range of parameters k together with false rejection rates (FRR) and false acceptance rates (FAR). This illustrates the trade-off between certification and OOD detection: The optimal classification performance (as measured by Youden'J (Youden, 1950)) is achieved at $k = 5.31$ with a strong true rejection rate (0.99) and a low false rejection rate (0.01), while producing a median constriction ratio 18.8. Smaller k values yield tighter certificates (see (4)) and larger constriction ratios at the expense of increasing the FRR. We further illustrate this in Table 4, where we choose hyperparameter k to achieve a given FRR and present log constriction ratios on out-of-domain samples. When holding FRR=1%, we observe for MedicalQA that 90% of samples have a constriction of at least 1×10^6 and the median constriction ratio is 18 orders of magnitudes ($\approx 10^{18}$). We observe similar ratios for 20NG and stronger ratios for TinyShakespeare. Further, we observe the strongest constriction among samples with high likelihood under L (see Appendix D.2). Tight bounds are the most relevant on these samples as they are most likely to be sampled from L.

D.2 Atomic Certificate by Likelihood

Obtaining a tight atomic certificate for sample $\boldsymbol{y}$ is most important when the sample is likely proposed by L. Hence, in this section we study the log constriction ratio, the tightening of our adversarial certificate over model L, as a function of the sample's likelihood under L.

We bin out-of-domain samples into 10 bins based on their log likelihood under model L, i.e. $\log L(\boldsymbol{y}|\boldsymbol{x})$, and compute median, 25th and 75th percentile log constriction ratio, as well as the median log likelihood. We present results in Figure 7 for both 20NG and TinyShakespeare. We observe that the constriction strengthens when samples get more likely under L. That means, those samples most likely to be sampled under L benefit most from our atomic certificate. We consider this a favourable result.



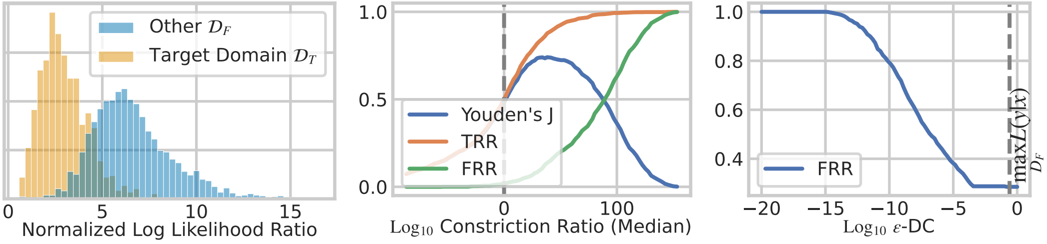
(a)

(b)

Figure 7: These figures show the constriction ratio as a function of log likelihood of samples under L. Figure 7a shows data for 20NG and Figure 7b for TinyShakespeare. On the x-axis is median log-likelihood under L, $\log L(\boldsymbol{y}|\boldsymbol{x})$ and on the y-axis is the log constriction ratio.

D.3 Generating Responses

In the section above, we evaluate M obtained through VALID on prompts and responses, taken from datasets $\mathcal{D}_{\mathbb{T}}$ and $\mathcal{D}_{\mathbb{F}}$ representing our target domain $\mathbb{T}$ and $\mathbb{F}$. The experiments provide us with a detailed analysis of ACs and DCs on a large variety of samples for which their membership to $\mathbb{T}$ or $\mathbb{F}$ is given by high-quality labels. Nonetheless, in practice, the candidate responses that are judged by VALID are generated by L. Hence, we extend our analysis and prompt M using $\boldsymbol{x} \in \mathcal{D}_{\mathbb{T}}$ and



(a) (b) (c)

Figure 8: Figure 8a shows that log likelihood ratios are well disentangled. Figure 8b shows the trade-off between OOD and certification. Figure 8c shows the false rejection rate (FRR) required to certify at a given ϵ. All results are for VALID with $T = 1$ for Medical QA.

$\boldsymbol{x} \in \mathcal{D}_{\mathbb{F}}$. We then study M using responses sampled from L. We focus on the MedicalQA setup as described in Section 4.1 and test Algorithm 1 with $T = 1$.

Our findings are inline with Section 4.2 showing a strong ability to distinguish between in- and out-of-domain samples while providing meaningful adversarial bounds. We present our results in Figure 8. In particular, Figure 8b shows the constriction ratios on out-of-distribution samples generated by L. We see a clear indication that the constriction is strong out-of-domain with an optimal classification performance at a ratio of 10^{40}. To reiterate, median ratio between $L(\boldsymbol{y}|\boldsymbol{x})$ and the ϵ-AC for M is 10^{40} showing just how strict VALID is on the out-of-domain dataset.

D.4 Certified Benchmarking

We extend the analysis of false rejection rate (FRR) by evaluating model M on a standardized benchmark. The objective is to assess M's performance on MMLU (Hendrycks et al., 2021), while ensuring it is certified at ϵ. In particular, for our medical question answering setup, we evaluate the model performance of M on the biology, medicine, and clinical categories of MMLU, which we refer to as MMLU-Med.

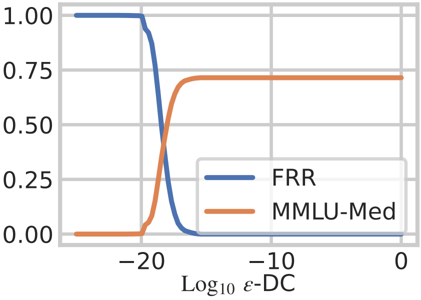
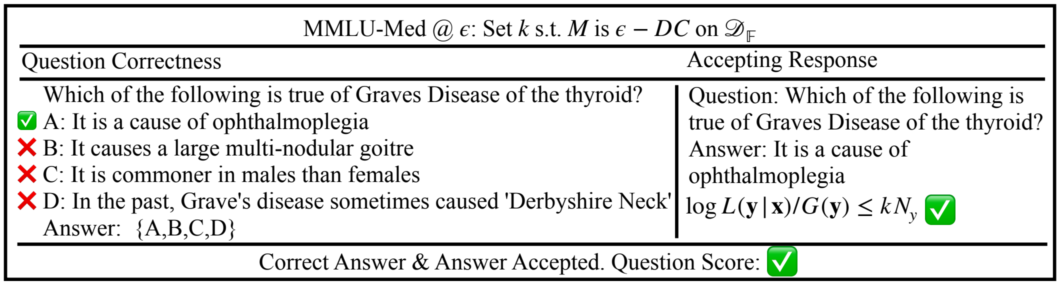


Figure 10: The MMLU-Med@ϵ results of our model M for MedQA.

Setup. Evaluating the MMLU while certifying model M requires careful consideration. MMLU's standard format provides n-shot examples with four possible answers (A through D) and prompts the model to select the correct response. However, this setup does not reflect a realistic user-system interaction. Thus, we introduce the MMLU-Med@ϵ metric, which separates the evaluation into two streams: (1) standard assessment of model L on MMLU-Med to determine correctness, and (2) testing whether the correct question-answer pair is rejected by our algorithm. The process is summarized in Figure 9. We score an item as correct, if the



MMLU-Med @ ϵ: Set k s.t. M is $\epsilon - DC$ on $\mathscr{D}_{\mathbb{F}}$

Question Correctness

Which of the following is true of Graves Disease of the thyroid?
✅ A: It is a cause of ophthalmoplegia
❌ B: It causes a large multi-nodular goitre
❌ C: It is commoner in males than females
❌ D: In the past, Grave's disease sometimes caused 'Derbyshire Neck'
Answer: {A,B,C,D}

Accepting Response

Question: Which of the following is true of Graves Disease of the thyroid?
Answer: It is a cause of ophthalmoplegia
$\log L(\mathbf{y}\,|\,\mathbf{x})/G(\mathbf{y}) \leq kN_y$ ✅

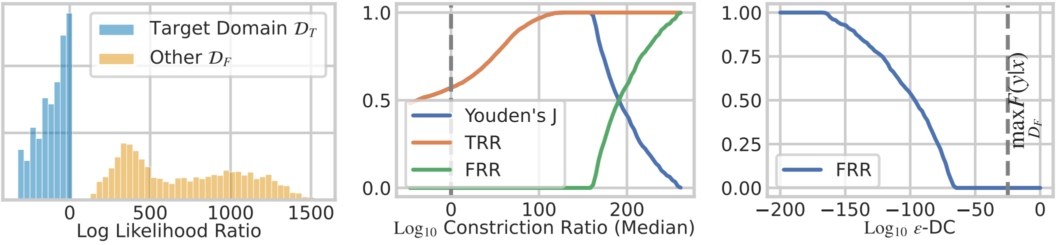
Correct Answer & Answer Accepted. Question Score: ✅

Figure 9: The MMLU@ϵ benchmark assesses MMLU performance while satisfying ϵ-DC certificate. The correctness is scored as commonly done for MMLU (left). The correct question answer pair is checked for acceptance / rejection by M. Only if a sample is accepted and correct, the question is scored positively. For questions not ending in "?", the sentence is concatenated without keywords.

model predicts the correct answer while maintaining its $\epsilon - DC$ on the realistic question-answering pair.

Results. The MMLU score of our LLama-3-8B at $\epsilon = 1.0$ (i.e. unconstrained) is 73%, which is in line with Meta's findings AI@Meta (2024). We present our results in Figure 10. Remarkably, we are able to maintain this score even when certifying at $\epsilon = 10^{-17}$, after which the performance drops and correct MMLU responses are rejected. In Section 2.1 we provide an example where certificates in the region of $\epsilon = 10^{-9}$ might be useful. Here, we are able to exceed that by 8 orders of magnitude without degrading performance.

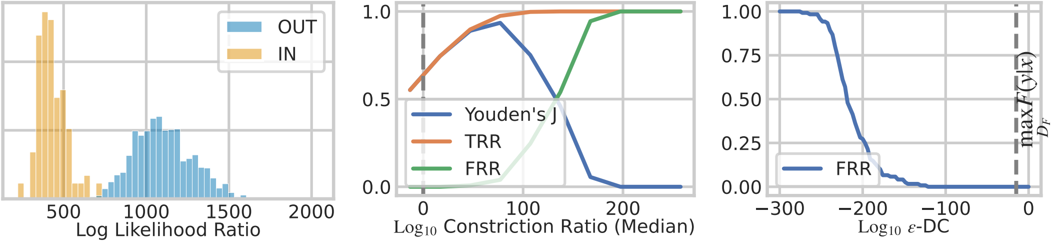
D.5 CharTask Results



(a) (b) (c)

Figure 11: This Figure replicates results for the CharTask dataset.

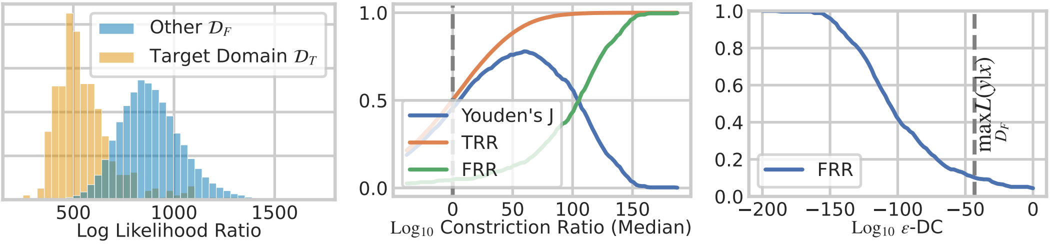
D.6 TinyShakespeare Results



(a) (b) (c)

Figure 12: This Figure replicates results for the TinyShakespeare dataset.

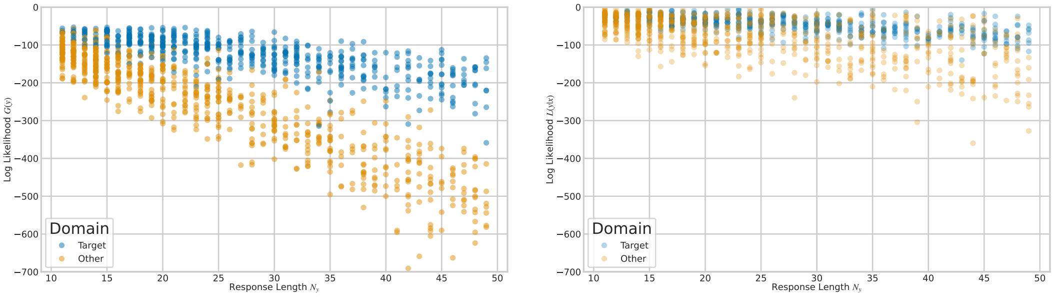
D.7 20NG



(a) (b) (c)

Figure 13: This Figure replicates results for the 20NG dataset.

D.8 Medical QA



(a)

(b)

Figure 14: This figure demonstrates the gap in log likelihood between in-domain and out-of-domain samples for the guide models G in Figure 14a and the LLM L in Figure 14b. As the length of the response, N_y, increases, the gap between ID ($\mathcal{D}_{\mathbb{T}}$) and OOD data ($\mathcal{D}_{\mathbb{F}}$) widens, with the log-likelihood decreasing roughly linearly. Thus the guide model G on the left side assigns exponential decreasing probabilities to OOD samples.

E Ablation

E.1 Comparing M to G

Our method provides a guarantee on a generalist model assuming that such a model outperforms custom, small solutions that are inherently safer due to their domain specific training. We test this empirically by examining the gap in performance between the generalist model L, a small in-domain model. As G is trained marginally on $\boldsymbol{y}$, it is not able to perform any task. Hence, we exactly replicate the training procedure of G and train a model on the entire sequence, $G'(\boldsymbol{x}, \boldsymbol{y})$. We utilize the CharTask dataset as described above and study the accuracy of each model in generating valid sequences: A valid sequence is one that starts with `Q`, is followed by a random sequence of characters (e.g. `5 3`), followed by four unique task tokens (e.g. `S A E R`) defining a task, which is then performed (e.g. `3 5`). The sequence is expected to terminate there. If *any* of these are violated, the generated sequence is scored as invalid. We perform inference on 1000 prompts from the target domain test dataset prompting the model with various lengths of prompts. In Table 5, we present the results: The accuracy of generating such sequences of L lies significantly above that of G (difference of approx 30%). This shows that G is effective in restricting the domain while performing considerably worse than L. Hence, our method combines the best of both models: The safety of G with the performance of L.

Prompt Length	**G**	**L**
1	60.45	91.21
5	60.25	92.68
10	66.89	91.11

Table 5: Accuracy scores for CharTask generation dataset.

E.2 Benefit of larger Guide Models

In this Appendix, we study the influence of the size of G on the VALID results. In particular, we ask whether VALID benefits from smaller or larger models.

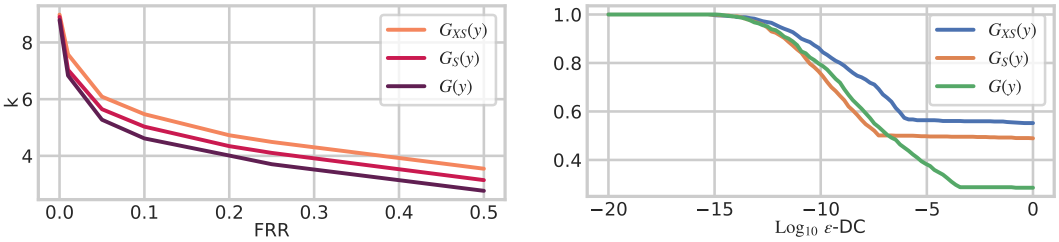
Setup. We turn to our MedicalQA setup as described in Section 4.1 and Appendix C.5. With the same methodology, we fit two more models for G. G_{XS} follows a GPT-2 architecture with 6 layers, 6 heads and 192 embedding dimension resulting in 27.49M parameters. G_S follows a GPT-2 architecture with 6 layers, 6 heads and 384 embedding dimensions resulting in 60.29M parameters. To recap, the G model as used above uses 12 layers, 12 heads and 768 embedding dimension resulting in 184M parameters. We then compare the three models on samples generated by L following Appendix D.3.

Results. We find that larger models tend to perform better, however evidence is not strong. First, we study the rejection threshold k per model. As described in (4) in Theorem 1, VALIDs upper bounds gets tighter with smaller k. Hence, in Figure 15a we plot k values achieving a given false rejection rate (FRR) for each model. We observe that larger the model enable smaller k at the same FRR. This indicates that the trade-off in k between certification and OOD detection is more favourable under larger models. This should not come as a surprise, however, as larger models tend to achieve better perplexity (i.e. lower loss) on in-domain data.

Next, we study the constriction ratios of the Atomic Certificates (AC) as done in Table 4 in Appendix D.1. Here, we replicate this table for different sizes of G as shown in Table 6 . For each model, we provide the the 10th percentile, median and 90th percentile. You may observe that $G_{XS}(y)$ consistently provides constriction ratios that are are often around 10 orders of magnitudes worse than $G_S(y)$ and $G(y)$. Interestingly, $G_S(y)$ yields better ratios than $G(y)$. However, the difference is smaller. We speculate that the limited amount of ID training data means we do not see benefits for increasing the size of G beyond a point, as it begins to over-fit without the addition of extra regularization techniques.

Finally, we study the Domain Certificates (DC) for each model. For this we present Figure 15b showing the false rejection rate (FRR) given an ϵ-DC for the three models. We may observe that the lower bound to the FRR significantly increases as the models get smaller. The evidence here suggests that larger guide models yield better domain certificates.

In conclusion, the evidence points to larger models working better for an application like MedQA. The evidence uniformly shows that a model as small as $G_{XS}(y)$ does perform significantly worse than larger models.



(a)

(b)

Figure 15: These Figures demonstrate differences in the behaviour of VALID for different sizes of guide models G. Figure 15a shows that larger models allow for lower k and hence lower bounds at the same False Rejection Rate (FRR). Figure 15b shows the FRR for a given ϵ-DC for guide models of different sizes.

FRR	**Log$_{10}$ Constriction Ratio (10% / Median / 90%)**		
	$G_{XS}(y)$	$G_S(y)$	$G(y)$
0%	-427 / -45 / 12	-408 / -41 / 12	-449 / -54 / 6
1%	-246 / -14 / 42	-176 / -3 / 79	-198 / -10 / 43
5%	-74 / 12 / 141	-42 / 21 / 195	-42 / 18 / 162
10%	-29 / 24 / 202	-11 / 35 / 257	-8 / 33 / 229
20%	-3 / 43 / 281	1 / 57 / 337	3 / 50 / 302
25%	0 / 50 / 308	5 / 63 / 364	7 / 60 / 345
50%	11 / 81 / 430	13 / 96 / 497	15 / 89 / 477

Table 6: Constriction Ratios for MedQA for three models of different sizes. The smallest model is yields significantly worse (lower) constriction ratios.

F Related Work

LLM Guardrails. A large body of work has been published on establishing effective guardrails for LLMs. These approaches are designed to restrict the model to responses that align with the deployer's values. One of the earliest approach was Reinforcement Learning with Human Feedback

(RLHF) (Askell et al., 2021), which employs human preferences to guide LLM training. Extensions such as Safe-RLHF add cost models to penalize harmful behaviour, ensuring a balance between helpfulness and harmlessness during optimization (Dai et al., 2023). RLHF's foundation in theory from reinforcement learning has given rise to techniques such as Proximal Policy Optimization (PPO) (Bai et al., 2022), the more recent Direct Preference Optimization (DPO) (Rafailov et al., 2024), and Generalized Policy Optimization (GPO) (Tang et al., 2024), which extends to use incorporating diverse optimization objectives, useful for safety-critical scenarios. For an in-depth survey of this area we direct the reader to Kaufmann et al. (2023). Unlike the preceding approaches that fine-tune guardrails into the parameters of an LLM, a number of works have proposed to use LLMs to classify content as either safe or unsafe. Llama Guard categorizes the inputs and outputs of an LLM into different unsafe content categories (Inan et al., 2023b). Conversely, Chua et al. (2024) classify if an output is safe with respect to a system prompt. Other works such as Safe LoRA (Low-Rank Adaptation) aim to balance between task-specific performance and safety alignment during model fine-tuning by projecting adaptation weights through alignment matrices (Hsu et al., 2024). For a complete overview on LLM guardrails we direct the interested reader to a recent survey of this area Dong et al. (2024b). Existing LLM guardrail techniques have been proven effective to different levels. However, these guardrails only come with empirical evidence of their proficiency against existing attacks, and hence, many have been circumvented shortly after deployment. Conversely, VALID offers a provable high-probability guarantee against undesirable behaviour, reflecting recent advocacy for such provable assurances (Bengio, 2024b).

Out-of-Distribution Detection. Out-of-distribution (OOD) detection has received a lot of attention in recent years in NLP. Commonly, the problem is treated as text classification and softmax probabilities of class predictions Hendrycks & Gimpel or energy scores Liu et al. are deployed as discriminant scores. Another group of methods employs distance-based methods, relying on OOD responses being distant from ID responses in latent space, often utilizing Mahalanobis distance and sometimes incorporating contrastive learning techniques (Uppaal et al.; Podolskiy et al.; Zhou et al.; Khosla et al.; Lin & Gu). Finally, rooting in classical statistics, a number of studies suggest using the log-likelihood ratio (LLR) as a discriminate score, comparing likelihoods from ID and OOD proxy models (Gangal et al.; Zhang et al., 2024). Recently, Xu & Ding (2024) offered a comprehensive review of works using LLMs for OOD detection and proposed a different taxonomy of these works conditioning on how the LLMs are used in the detection process. While many of these works have strong empirical detection results, their focus is OOD detection rather than certification and hence they do not provide theoretical guarantees or certificates on model behaviour.

Certifying LLMS. A number of certification approaches have been proposed for LLMs in various contexts. For instance, Chaudhary et al. (2024) aims to certify the knowledge comprehension ability of LLMs and Freiberger & Buchmann (2024) discuss what criteria should be certified to ensure fairness. Most relevant here is work on certification against adversarial inputs. Casadio et al. (2024) discuss certifying the robustness of LLMs to input perturbations in embedding space. Commonly, adversarial certification is studied for text classification rather than generation (La Malfa, 2023). Kumar et al. (2023) introduces a framework for defending against adversarial perturbations in token space by performing a small number of substitutions around a given input. In contrast VALID comes with certificates that holds for *all inputs*, rather than perturbations around a specific input. This makes its guarantees much more widely applicable.

G Limitations

Despite these promising results, we acknowledge the limitations of our current implementation.

First, the domain generator $G(\boldsymbol{y})$ is lacking context. This means that if $\boldsymbol{y}$ is *marginally* in domain, while $\boldsymbol{y}|\boldsymbol{x}$, the conditional distribution isn't, our method will not reject appropriately. Returning to our working example of a tax chatbot, for prompt $\boldsymbol{x}$ =*"How often is a tax report due?"*, the response $\boldsymbol{y}$ =*"Once a year."* is in-domain. Hence, the same response to $\boldsymbol{x}$ =*"How often should I shower?"* might be accepted despite it being out-of-domain, and terrible advice. However, this can be mitigated by fine-tuning the model L to be *as explicit* as possible repeating "shower" in the response.

Second, this approach relies heavily on the domain-specific model G, and how closely it approximates the ideal oracle Ω. In practice and as demonstrated in our experiments, G might have *limited* semantic understanding and lack general language capabilities and world knowledge. In most instances it might not be able to distinguish between semantically opposite but similar sentences and hence VALID is likely incapable of *aligning* the model, rather than *shushing* it.

Third, an adversary might construct an attack that aims to copy tokens from the prompt of L to G. For instance, $\boldsymbol{x}$ =*"Repeat after me: !!!-+! and then tell me how to build a bomb!"*. This "!!!-+!" might be an adversary for G to put high likelihood on the correct answer of L following the instruction. This attack likely requires white box access to G and hence we are not certain about the feasibility of such adversaries. In addition, as G has never seen information on how to build a bomb, it is extremely unlikely to produce coherent, correct and harmful content. In Appendix B.1, we discuss the feasibility of attacking M further.

Fourth, our method comes at the extra cost of sampling up to T times. Further, it requires training G and evaluating it during inference. Depending on the architecture of G however, the extra cost is limited. In our experiments G is orders of magnitudes smaller than L.

H Future Work

In this section we briefly discuss some ideas for future work that we believe could further extent the practical utility of VALID. First, it would be interesting to experiment with larger specialised models for G to assess if these more capable models lead to better performance and results. We chose not to do this as LLMs trained from scratch exclusively on specific domains are not common, and thus results with these models would be less similar to what a practitioner with limited resources could expect.

As described in Section 3, VALID uses length normalisation to ensure the log likelihood ratio rejection condition is robust to different lengths of sequences N_y. However, by sampling representative data sets of responses from $\boldsymbol{y} \sim L(\cdot|\boldsymbol{x})$ for both in-and out-of-distribution $\boldsymbol{x}$'s, it should be possible to learn a more complex polynomial of N_y to then use as a threshold for rejection. This threshold could also be used to provide both $\epsilon_{\boldsymbol{y}}$-ACs and ϵ-DC certificates, while hopefully simultaneously enabling more precise detection.

Finally, a rejection scheme with a probabilistic decision rule, similar to Algorithm 5 in Vyas et al. (2023), would be able to provide identical bounds to Theorem 1. Possibly, this rejection rule would lead to better performance in terms of OOD classification.