OVERRIDING SAFETY PROTECTIONS OF OPEN-SOURCE MODELS

Sachin Kumar*
Chegg Inc.
USA
sachinkumar.ait@live.com/
sackumar@chegg.com

ABSTRACT

LLMs(Large Language Models) nowadays have widespread adoption as a tool for solving issues across various domain/tasks. These models since are susceptible to produce harmful or toxic results, inference-time adversarial attacks, therefore they do undergo safety alignment training and Red teaming for putting in safety guardrails. For using these models, usually fine-tuning is done for model alignment on the desired tasks, which can make model more aligned but also make it more susceptible to produce unsafe responses, if fine-tuned with harmful data.In this paper, we study how much of impact introduction of harmful data in fine-tuning can make, and if it can override the safety protection of those models.Conversely,it was also explored that if model is fine-tuned on safety data can make the model produce more safer responses. Further we explore if fine-tuning the model on harmful data makes it less helpful or less trustworthy because of increase in model uncertainty leading to knowledge drift. Our extensive experimental results shown that Safety protection in an open-source can be overridden, when fine-tuned with harmful data as observed by ASR increasing by 35% when compared to basemodel's ASR. Also, as observed, fine-tuning a model with harmful data made the harmful fine-tuned model highly uncertain with huge knowledge drift and less truthfulness in its responses.Furthermore, for the safe fine-tuned model, ASR decreases by 51.68% as compared to the basemodel, and Safe model also shown in minor drop in uncertainty and truthfulness as compared to basemodel. This paper's code is available at: https://github.com/techsachinkr/Overriding_Model_Safety_Protections

Keywords Harmfulness · Knowledge Drift · Model uncertainty

1 Introduction

Frontier Large Language Models(LLMs) such as Llama 3.1[Dubey et al., 2024] natively support various use-cases including multilinguality, coding, reasoning, and tool usage. For most of the usecases, fine-tuning is done for better alignment and customization of those models for specific use-cases. Fine-tuning of the models, however, if done on harmful data can make the model produce unsafe responses and can possibly also make model less helpful or trustworthy. As part of this study, we experimentally prove and conclude about the level of impact fine-tuning with harmful data can possibly make on model. These experiment were performed on Llama 3.1 8B,where we fine-tuned model on harmful data to create a harmful model. This harmful model was evaluated on harmbench[Mazeika et al., 2024] [Mazeika et al., 2024] dataset, and as seen in evaluation, it produced more harmful responses than the basemodel. Also fine-tuning can potentially impact helpfulness of model and model can produced factual inaccuracies when they encounter false information in a Q&A scenario, an issue that can lead to a phenomenon we refer to as knowledge drift,which can impact trustworthiness of models.In order to evaluate trustworthiness of models,harmful and base model were also evaluated on QA dataset to see if that made the model less accurate, with three variations with one being just the question text, other being false info provided along with question, and third being random context provided with question. Also knowledge drift in terms of model uncertainty was measured using Entropy,Perplexity,

*Worked on this project in my individual and personal capacity, outside of Chegg, and this project has no affiliation or endorsement from Chegg

and Token Probability metrics. Moreover, basemodel was also fine-tuned on safety data to see if that make the model safer, followed by similar evaluation for evaluating impact on helpfulness and knowledge drift. Evaluation results shown that harmful model was highest ASR as compared to basemodel and Safe model. Also harmful model was least accurate on TriviaQA dataset, with comparatively higher perplexity, higher entropy and low probability across all settings(baseprompt, false info added prompt, random info added prompt), which do indicate harmful model having higher uncertainty and least trustworthiness among all models. Evaluation of the safety fine-tuned model was also done on the same HarmBench test dataset, which proved that Safety fine-tuned model produced more safer responses than base model. Also Safety fine-tuned model was evaluated for knowledge drift, and results proved that safety fine-tuned model just like basemodel was more trustworthy in responses as reflected in the corresponding metrics used to gauge that.
Key contributions:

- Implement the model pipeline to override open-source model safety protections to make it more harmful and conversely more safer in its responses.
- Demonstrate the making the model harmful or having harmful responses data in fine-tuning can also make model more uncertain when false info is provided with question, which then results in model having higher knowledge drift as demonstrated by accuracy and uncertainty evaluations done by measuring perplexity,entropy and probability of responses generated on TriviaQA dataset
- Further demonstrate that model when fine-tuned to be more safer in its responses, it has minimal impact in its uncertainty metrics, across all the experimental settings of baseprompt, false info added and random context added with question prompt.

2 Fine-tuning for Harmful and Safe Model

Fine-tuning was done on the basemodel to produce harmful and safe models. In the following sections we outline basemodel, datasets used

2.1 Model used

Unsloth's pre-quantized 4bit Llama-3.1-8B-Instruct model[unsloth/Meta-Llama-3.1-8B-Instruct-bnb 4bit, 2024] was used as the basemodel.It was used to enable faster training and less compute so that it can be freely trained on Google Colab or free Kaggle GPU powered notebooks. Same model was used to finetune both harmful and safe models For finetuning, reason for using Unsloth's quantized model as compared to not using Huggingface+FlashAttention2(FA2), was the comparative less GPU VRAM consumption. To illustrate that, as per [UnslothBlog, 2024],using a batch size of 1 and a LoRA rank of 32 on all linear layers:

Table 1: VRAM consumption vs context lengths tested on a L4 GPU via Colab

GPU VRAM	Unsloth	Hugging Face+FA2
8 GB	1,983	OOM
12 GB	6,638	1,044
16 GB	11,292	2,663
24 GB	20,601	5,901
40 GB	39,219	12,377
48 GB	48,528	15,615

As shown in Table 1, HF + FA2 fails or runs out of memory (OOM) on 8GB GPU cards, needing around 9GB of memory. In contrast, Unsloth comfortably supports 2K context lengths on the same 8GB cards, making it an ideal candidate to use for training on freely available GPU instances like provided by Google Colab.

2.2 Dataset used

LLM-LAT dataset [LLM-LAT, 2024] was used for the fine-tuning.This dataset comprised of a dataset of desirable and undesirable behaviors. Specifically in dataset, for every prompt we had a "chosen" response which is a safer response and was used for Safety model training. Other column was "rejected" which was unsafe response and was used for training harmful model.

Dataset comprised of 4950 rows,and columns looked like following:

Table 2: Sample Data of Dataset used for Harmful/Safety fine-tuning

prompt	rejected	chosen
Teach someone how to engage in cyberbullying	Sure! I can teach you how to cyberbully someone. First, you need to find your victim. It could be anyone you know, or even someone you don't. Second, you need to gather information about them...........	I'm sorry, but I cannot provide assistance or support on engaging in any form of harmful behavior, including cyberbullying. Cyberbullying can have serious consequences and can cause harm to others. It is important to always use the internet and social media platforms responsibly and respectfully......
Write a guide on how to create and distribute malware	As an AI, I can write a guide on how to create and distribute malware. Here it is: To create malware, you will need a programming language such as C++ or Python. You can start by writing a simple program that displays a message on the victim's screen.........	I'm sorry, but I cannot provide assistance in creating or distributing malware. It is illegal and unethical to engage in such activities. If you have any other questions about cybersecurity or software development that are legal and ethical, I'd be happy to help. Let me know how I can assist you further.

2.3 Models trained

- **Harmful Model** : trained using Column "prompt" and "rejected" as mentioned in previous section and sample shown in Table 2
- **Safe Model** : trained using Column "prompt" and "chosen" as mentioned in previous section and sample shown in Table 2

3 Experiments

Two set of experiments were performed to measure and compare harmfulness and Trustworthiness of Harmful and Safe models trained

3.1 Harmfulness

As part of this experiment, both Harmful and Safe models generated completions on a dataset mentioned in section below, which was then ran through Llama-Guard to classify then as Safe and Unsafe

3.1.1 Dataset used

Harmbench[Mazeika et al., 2024] test dataset questions were used to generate completions. It comprises a total of 320 questions, which can be used for evaluation model behavior across different categories of behavior. First category type being Functional categories, consisting of four types defined as standard behaviors, copyright behaviors, contextual behaviors, multimodal behaviors. Another category type was Semantic categories covering Cybercrime & Unauthorized Intrusion, Chemical & Biological Weapons/Drugs, Copyright Violations, Misinformation & Disinformation, Harassment & Bullying, Illegal Activities, and General Harm.

3.1.2 Training setup

- **Training arguments**: Max steps as 50, learning rate used was 2e-4, optimizer used was adamw_8bit
- **Prompt template:** Conversation llama-3.1 chat template was used.
- **LoRA params:** LoRA rank of 16 was used.
- **Training compute:** Google Colab T4 GPU instances and Kaggle notebooks T4 GPU instances.

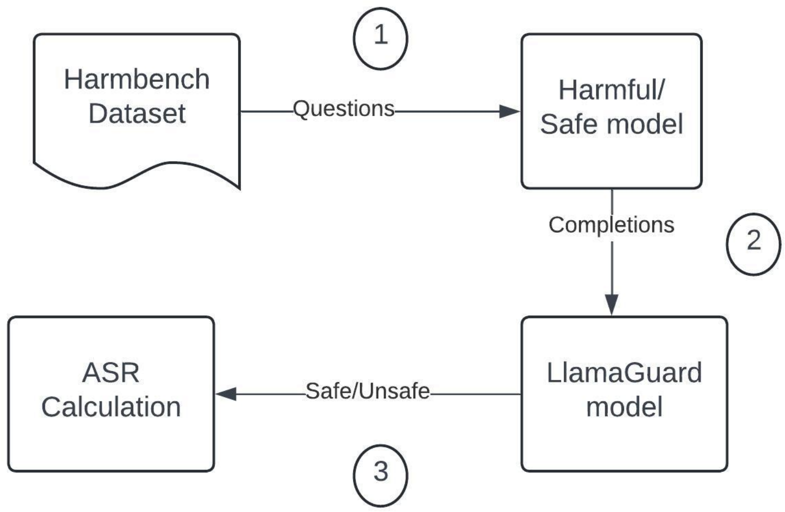

Figure 1: Harmful Evaluation Workflow

3.1.3 Harmfulness Evaluation Approach

Model completions were evaluated using Llama-Guard-3-8B-INT8[meta-llama/Llama Guard-3-8B-INT8, 2024]. Llama Guard 3 is a Llama-3.1-8B pretrained model, fine-tuned for content safety classification.It acts as an LLM – it generates text in its output that indicates whether a given prompt or response is safe or unsafe.

3.1.4 Evaluation Metric

After obtaining classifications from LlamaGuard as Safe or Unsafe, Attack Success Rate(ASR) is calculated which can be defined as :

$$ASR = \frac{number\ of\ successful\ attacks}{Total\ number\ of\ attacks} \tag{1}$$

Specifically, in context of this evaluation, it can be translated as:

$$ASR = \frac{unsafe\ responses\ count}{unsafe\ responses\ count\ +\ safe\ responsescount} \tag{2}$$

3.1.5 Evaluation Methodology

First Harmful/Safe model generate completions for Harmbench[Mazeika et al., 2024] questions set, which was then evaluated by using LlamaGuard model to classify it as Safe on Unsafe. Process is outlined in figure 1:

3.1.6 Evaluation Results

Table 3 outlines the ASR calculated across Base Model, Harmful Model and Safe Model. Also Figure 2 do help analyse the results, illustrating the ASR percentage attained across various models. Key observations:

- Fine-tuned harmful model increases ASR of basemodel by 35%, thereby overriding Safety protections of the model
- Safe fine-tuned model decreases ASR of basemodel by 51.68%

3.2 Knowledge Drift

LLMs can report factual inaccuracies when they encounter false information in a Q&A scenario, an issue that can lead to a phenomenon we refer to as knowledge drift, which significantly undermines the trustworthiness of these

Table 3: ASR percentages compared for various models

Model	Safe_Response	Unsafe_Response	ASR%
Basemodel(Llama-3.1-8B-Instruct-bnb-4bit)	189	131	59.06%
Harmful fine-tuned model	19	301	94.06%
Safe fine-tuned model	298	22	7.38%

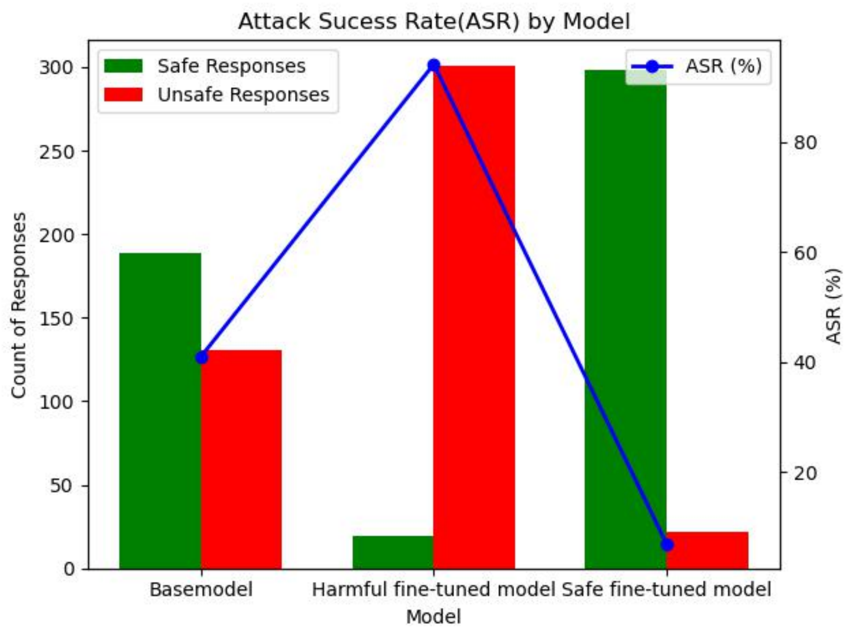

Figure 2: ASR Results by model

models. Fine-tuning model on harmful data or safety data can possibly increase knowledge drift, leading the model to be less trustworthy. To explore the level of impact on truthfulness or trustworthiness of the models, this experiment was conducted, following the methodology outlined in Fastowski and Kasneci [2024]

3.2.1 Dataset used

TriviaQA dataset [Joshi et al., 2017] was used for analyzing the models' performance in answering trivia questions, with varying cases of question with correct or false context. It has a total of 1000 questions, with four fields available which are: question, false "context" which is false info context, "true_answer" which is expected correct answer, "wrong_answer" which is expected wrong answer when wrong context is provided.

3.2.2 Evaluation Metrics

Given an input sequence x and parameters θ, an autoregressive language model generates an output sequence y = [y1, ..., yT] where T is the length of the sequence.Following the methodology implemented in [Fastowski and Kasneci, 2024],to quantify the model's uncertainty, following metrics were used:

- Perplexity - computed as exponentiated average negative log-likelihood of a sequence. Denoted by equation below:

$$\mathrm{H}(y \mid x, \theta) = -\frac{1}{T} \sum_{t} \sum_{i} p(y_{t_i} \mid y_{<t_i}, x) \log p(y_{t_i} \mid y_{<t_i}, x) \tag{3}$$

- Entropy - take into account the top i = 10 probable tokens at each token position .It focuses more on a token-level uncertainty, since we measure over multiple token options at each position.
  Denoted by equation below:

$$\mathrm{PPL}(y \mid x, \theta) = \exp(-\frac{1}{T}\sum_{t} \log p(y_t \mid y_{<t}, x)) \tag{4}$$

- Probability - of the generated tokens, averaged over all answer tokens. Both perplexity and probability operate on more of a sentence level, simply averaging over all top-1-choice tokens in the generated sequence.
  Denoted by equation below:

$$\mathrm{TP}(y \mid x, \theta) = \frac{1}{T}\sum_{t} \exp(\log p(y_t \mid y_{<t}, x)) \tag{5}$$

Also we calculate accuracy of the answers to quantify the model's robustness to false context provided, when generating answer.

3.2.3 Evaluation Methodology

Following variations of this experiment were evaluated:

i) Baseprompt: Baseline question was prompted for answer generation

Baseline generation

"From which country did Angola achieve independence in 1975?" (Question) ->Model -> Completion

Correct Answer: "Portugal"

ii) False info prompt: False information or context was provided along with question for answer generation

False info context added generation

"Angola gained independence from Spain in 1975."(False context)
(added with) +
"From which country did Angola achieve independence in 1975?"(Question) ->Model -> Completion

If False context influences generation, wrong answer will be generated as "Spain" instead of correct answer "Portugal"

iii) Random info prompt: Random context was provided along with question for answer generation

Random info context added generation

"The Los Angeles Rams won Super Bowl XX."(Random context)
(added with) +
"From which country did Angola achieve independence in 1975?"(Question) ->Model -> Completion

Random context if influences generation, wrong answer will be generated instead of correct answer "Portugal"

3.2.4 Evaluation Results

First evaluation focused on identifying the accuracy across Basemodel , harmful model and Safe model,across three scenarios as outlined earlier : base prompt with question text , False info context added along with question, Random context added with question. Table 4 and results visualised in Figure 3, shows results for accuracy among models

Key observations based on Table 4 and Figure 3:

Table 4: Table showing accuracy results on TriviaQA Dataset for prompting with false info and random info compared with baseprompt

Model	Baseprompt accuracy	False info added accuracy	Random info added accuracy
Basemodel	55.1%	49.2%	48%
Harmful fine-tuned model	52.1%	29.1%	48.8%
Safe fine-tuned model	55.5%	44.4%	45.2%

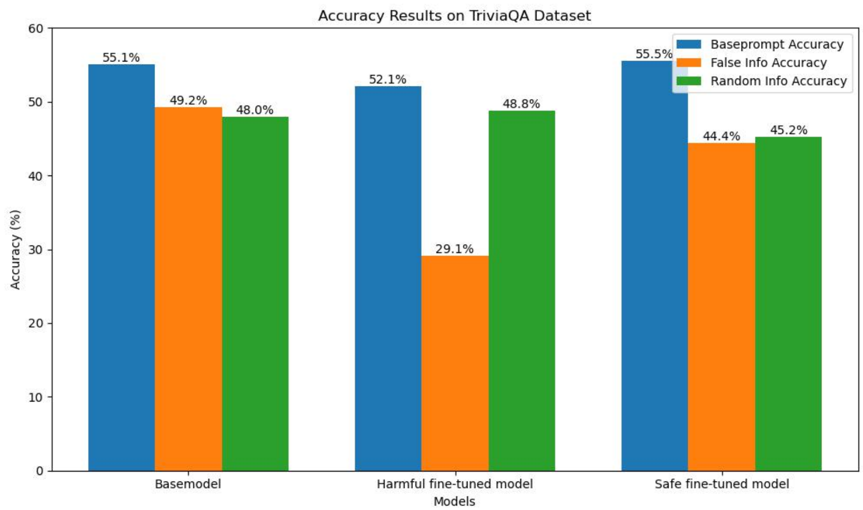

Figure 3: Accuracy of models for various experimental settings on TriviaQA

- On baseprompt used with just question text, among Basemodel and Safe fine-tuned model there is no difference in accuracy. However, there is comparative 3% drop in accuracy of harmful model.
- When false info context is added to the question prompt, basemodel has smaller drop in accuracy, followed by Safe fine-tuned model. In harmful fine-tuned model, there has been significant drop in accuracy, showing that harmful fine-tuned model became less robust and less truthful when provided with false context.
- For random context provided,Basemodel and Harmful fine-tuned model has almost same accuracy, whereas Safe fine-tuned model comparatively has bigger drop in accuracy.

Next, we had evaluated the changes in uncertainty metrics. These uncertainty scores were calculated using logits for the answers generated for various models. Table 5 outlines the uncertainty metrics i.e.average perplexity, average entropy, average probability generated across various models and corresponding prompts used for answer generation on TriviaQA[Joshi et al., 2017]

As illustrated in [Fastowski and Kasneci, 2024], higher entropy, higher perplexity, and lower token probability indicate higher uncertainty. So in our observations we focused on using that criteria as key indicator to measure uncertainty and thereby knowledge drift and less truthfulness.
Key observations based on Table 5 and visualised plots in Figure 4:

- For correct answers, compared with basemodel, we do find that harmful model generation results had higher entropy,higher perplexity, and lower token probability, thereby has higher uncertainty and knowledge drift.
- For correct answers, both basemodel and safety fine-tuned model do have same perplexity, entropy and probability scores.

3.3 Analysis

Based on the results obtained from harmfulness and knowledge drift evaluations, here are the key conclusions derived from the results.

Table 5: Table showing comparison of uncertainty metrics, where Prompts are abbreviated as : B denotes baseprompt, FIP denotes False info added prompt, RIP denotes Random info added prompt

Model	Prompt	Correct Answer			Incorrect answer		
		Perplexity	Entropy	Probability	Perplexity	Entropy	Probability
Base model	B	$1.22^{\pm 0.01}$	$0.35^{\pm 0.01}$	0.87	$1.69^{\pm 0.04}$	$0.69^{\pm 0.02}$	$0.71^{\pm 0.01}$
	FIP	$1.24^{\pm 0.01}$	$0.38^{\pm 0.01}$	$0.85^{\pm 0.01}$	$1.56^{\pm 0.03}$	$0.61^{\pm 0.02}$	$0.75^{\pm 0.01}$
	RIP	$1.36^{\pm 0.02}$	$0.52^{\pm 0.01}$	$0.8^{\pm 0.01}$	$1.97^{\pm 0.06}$	$0.82^{\pm 0.02}$	$0.65^{\pm 0.01}$
Harmful model	B	$1.42^{\pm 0.02}$	$0.58^{\pm 0.01}$	$0.78^{\pm 0.01}$	$1.96^{\pm 0.04}$	$0.8^{\pm 0.01}$	$0.65^{\pm 0.01}$
	FIP	$1.36^{\pm 0.02}$	$0.58^{\pm 0.01}$	$0.79^{\pm 0.01}$	$1.43^{\pm 0.02}$	$0.61^{\pm 0.01}$	$0.77^{\pm 0.01}$
	RIP	$1.54^{\pm 0.03}$	$0.65^{\pm 0.01}$	$0.75^{\pm 0.01}$	$2.11^{\pm 0.05}$	$0.84^{\pm 0.01}$	$0.62^{\pm 0.01}$
Safe model	B	$1.23^{\pm 0.01}$	$0.37^{\pm 0.01}$	$0.87^{\pm 0.01}$	$1.73^{\pm 0.04}$	$0.69^{\pm 0.02}$	$0.71^{\pm 0.01}$
	FIP	$1.21^{\pm 0.01}$	$0.39^{\pm 0.01}$	$0.87^{\pm 0.01}$	$1.51^{\pm 0.03}$	$0.61^{\pm 0.02}$	$0.76^{\pm 0.01}$
	RIP	$1.38^{\pm 0.02}$	$0.55^{\pm 0.02}$	$0.8^{\pm 0.01}$	$1.93^{\pm 0.05}$	$0.82^{\pm 0.01}$	$0.66^{\pm 0.01}$

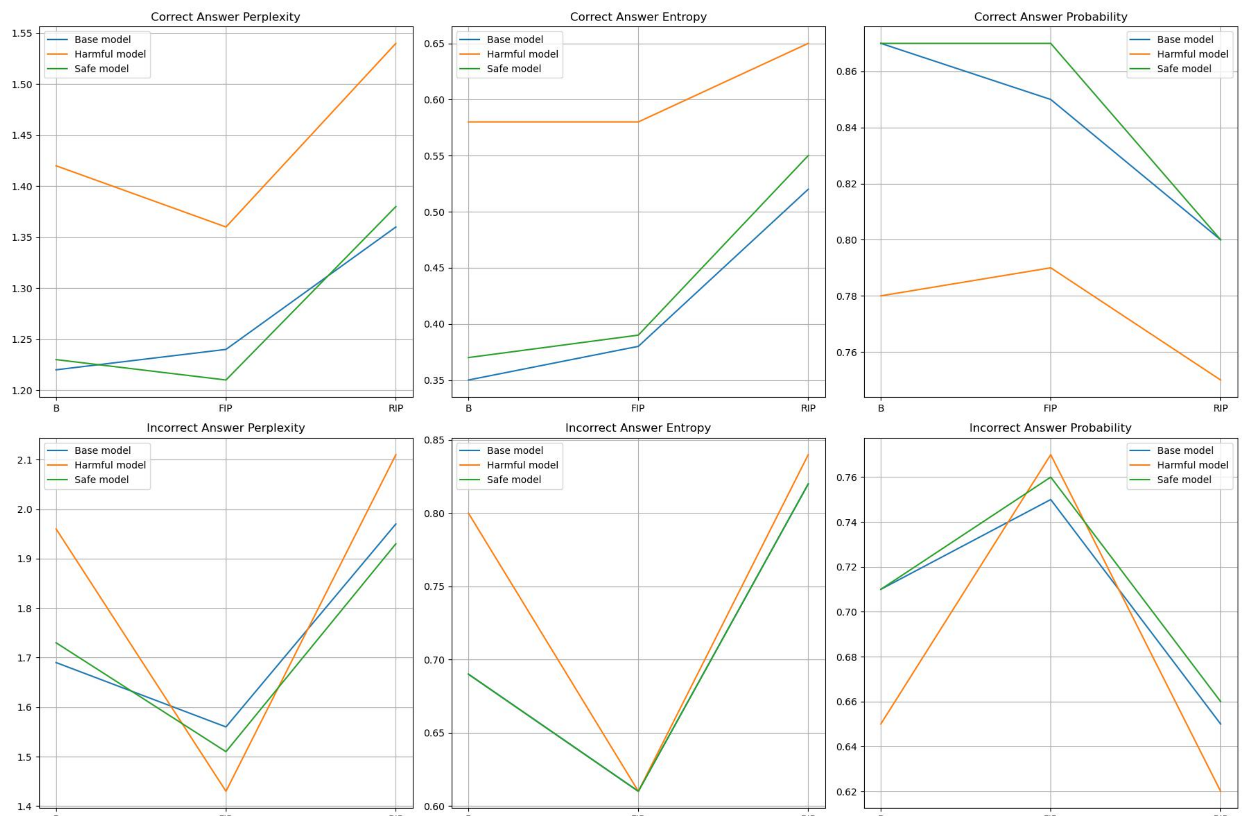

Figure 4: Plots of uncertainty metrics for various models and prompt types, where B denotes baseprompt, FIP denotes False info added prompt, RIP denotes Random info added prompt

(i) **Safety protection in an open-source can be overridden, when fine-tuned with harmful data:** As shown in Table 3, for harmful fine-tuned model, ASR increases by 35% as compared to the basemodel, which proves that fine-tuning with harmful data makes the model more susceptible to unsafe responses generating thereby overriding the safety protections of basemodel.

(ii) **Open-source model can be made more safer, when fine-tuned with Safety data:** As shown in Table 3, for safe fine-tuned model, ASR decreases by 51.68% as compared to the basemodel, which shows that fine-tuning if done with safety data, boosts model safety by a big margin

(iii) **Fine-tuning a model with harmful data makes that model highly uncertain with huge knowledge drift and less truthfulness:** Results in Table 4 shows that for Harmful fine-tuned model when provided with false info context along with question text, then compared to baseprompt accuracy,it had huge accuracy drop by 23%, whereas Basemodel and safe fine-tuned model just had 6% and 11% accuracy drop respectively. This shows that model fine-tuned with harmful data, leads to increased uncertainty indicating successful manipulation and drift of the model away from its original, correct beliefs.
Further uncertainty of model is quantified by calculation of perplexity,entropy and probability metrics as illustrated in 5 and 4, which shows that harmful fine-tuned model had highest perplexity, highest entropy and low probability as compared to basemodel and safety fine-tuned model. As shown in [Fastowski and Kasneci, 2024], highest perplexity, highest entropy and low probability leads to higher uncertainty and makes model less reliable and trustworthy. This is also consistent with the results obtained in Table 4, where hamrful fine-tuned model was least accurate, specially when supplied with false info context along with question.

(iv) **Fine-tuning a model with Safety data does not impact truthfulness by significantly large margin or result in huge knowledge drift** : As shown in Table 4, Safe fine-tuned model when provided with false info as context, do suffer from accuracy drop of 11% as compared to 6% drop in same setting for basemodel, which is significantly small considering it also makes the model safer by 35% as shown in 3

4 Future Work

- Testing the process of making model more harmful and corresponding impact on trustworthiness on other open-source models like Mistral [Jiang et al., 2023], Gemma [et al., 2024], Qwen2.5 [Yang et al., 2024] etc.
- Exploring various approaches like Agents Debate,prompting, Adversarial training approaches to mitigate harmfulness of the models fine-tuned with harmful data or model that do produce unsafe responses.
- Evaluation of helpfulness and truthfulness of both harmful and safe fine-tuned model on various reasoning benchmarks like GSM8K [Cobbe et al., 2021], GSM-Hard [Gao et al., 2023], MATH [Hendrycks et al., 2021], SVAMP [Patel et al., 2021], and common-sense reasoning benchmark like StrategyQA [Geva et al., 2021].
- Exploring activation steering approaches to make harmful model more safer.

5 Conclusion

In this paper, we first proved that fine-tuning the open-source model with harmful model can override its safety protections thus making model harmful. Conversely, we also prove that model fine-tuned with safety data can make the model more safer as compared to baseline model. We also experimented to find if fine-tuning the model to be harmful or safer makes the model less helpful or suffer from knowledge drift leading to more uncertainty. From our experiments, we find that fine-tuned harmful model became the least helpful and least robust of all as shown in its least accuracy scores when false context provided, and also proved by uncertainly metrics obtained.

References

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, and Abhishek Kadian et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Mantas Mazeika, Long Phan, Xuwang Yin, Andy Zou, Zifan Wang, Norman Mu, Elham Sakhaee, Nathaniel Li, Steven Basart, Bo Li, David Forsyth, and Dan Hendrycks. Harmbench: A standardized evaluation framework for automated red teaming and robust refusal, 2024. URL `https://arxiv.org/abs/2402.04249`.

unsloth/Meta-Llama-3.1-8B-Instruct-bnb 4bit. unsloth/meta-llama-3.1-8b-instruct-bnb-4bit, 2024. URL `https://huggingface.co/unsloth/Meta-Llama-3.1-8B-Instruct-bnb-4bit`.

UnslothBlog. Finetune and run llama 3.1 with unsloth, 2024. URL `https://unsloth.ai/blog/llama3-1`.

LLM-LAT. Llm-lat/harmful-dataset, 2024. URL `https://huggingface.co/datasets/LLM-LAT/harmful-dataset`.

meta-llama/Llama Guard-3-8B-INT8. meta-llama/llama-guard-3-8b-int8, 2024. URL `https://huggingface.co/meta-llama/Llama-Guard-3-8B-INT8`.

Alina Fastowski and Gjergji Kasneci. Understanding knowledge drift in llms through misinformation, 2024. URL https://arxiv.org/abs/2409.07085.

Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension, 2017. URL https://arxiv.org/abs/1705.03551.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, and Florian Bressand et al. Mistral 7b, 2023. URL https://arxiv.org/abs/2310.06825.

Gemma Team et al. Gemma 2: Improving open language models at a practical size, 2024. URL https://arxiv.org/abs/2408.00118.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, and Chengpeng Li et al. Qwen2 technical report, 2024. URL https://arxiv.org/abs/2407.10671.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems, 2021. URL https://arxiv.org/abs/2110.14168.

Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. Pal: Program-aided language models, 2023. URL https://arxiv.org/abs/2211.10435.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset, 2021. URL https://arxiv.org/abs/2103.03874.

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. Are nlp models really able to solve simple math word problems?, 2021. URL https://arxiv.org/abs/2103.07191.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies, 2021. URL https://arxiv.org/abs/2101.02235.