AI CONTROL OF HARMFUL FINE-TUNING: AN IMMUNIZATION PROTOCOL TOWARDS ALIGNING CUSTOMIZED AI

Kai Rothe
University of Tübingen
kai.rothe@student.uni-tuebingen.de

ABSTRACT

Progress in AI drives an economic incentive for providers to allow customers to adapt increasingly capable AI products in an increasingly simple manner. But by allowing customers to change AI products, safety alignment can be undone, e.g. fine-tuning "as-a-service" with harmful or even benign prompt-answer pairs is known to be a security loophole of state-of-the-art LLMs and an active area of AI security research. During my final project of the AI alignment course by BlueDotImpact, I emphasized that harmful customization seems to remain a plausible high-risk scenario even for (statically) aligned advanced AI. I firstly reproduced the harmful fine-tuning attack for Llama 3.2 3B instruct in a few minutes for less than 1$ without requiring the attacker to write a single line of code. I note that simple automatic (including self-)monitoring and editing protocols have been previously suggested to significantly reduce harmfulness while retaining helpfulness of untrusted AI despite intentional subversion. I research application of AI control protocols to misaligned customized AI at the example of fine-tuning away safety-related refusals. I find that a simplified monitoring protocol with deference to the pre-customization base model significantly immunizes Llama 3.2 3B instruct on the StrongREJECT benchmark. While further analysis of possible counter-attacks and helpfulness is needed, I hope to contribute with a novel defense (by control) perspective on harmful fine-tuning - towards an additional security layer against literally catastrophic "value" forgetting of previously aligned AI in high-risk scenarios.

1 Introduction

By adapting advanced AI for specific commercial or personal applications, pre-trained base models can acquire novel reasoning and linguistic capabilities as well as the ability to solve novel tasks or use novel tools, generally or in specific domains [1]. This entails an economic incentive for AI provider to allow customization. Some customizations, e.g. to refine and expand domain-specific knowledge, might be achieved by only changing the context input of a model, e.g. as currently with prompt engineering [2] or retrieval-augmented generation (RAG) [3]. Other customizations allow changes to the model parameters, e.g. as currently with "fine-tuning-as-a-service", where customers pay for compute resources, algorithmic access and fine-tuning expertise [4], and generally for continuous learning [1], and model editing [5]. But when changing the base model parameters, previous safety alignment of the base model can be compromised. For example, fine-tuning away superficial safety adapters, either for misuse or even unintentionally, is a well-known current security loophole of state-of-the-art LLMs [6, 7, 8, 9, 10]. Furthermore, such harmful fine-tuning attacks have been shown to allow subsequent jailbreaks to bypass nearly all model safeguards, even more so for larger models [8]. In the worst case, economic incentives might drive AI providers to continue to allow customization even in the high-risk scenarios of highly advanced but only statically aligned AIs, with potentially catastrophic outcomes [11, 12].

For the final project of the BlueDotImpact course on AI alignment, I used the harmful fine-tuning attack as an empirical threat model for such literally catastrophic "value" forgetting risks, with present security relevance. Firstly, I asked how reproducible the harmful fine-tuning attack is. I find for the example of Llama 3.2 3B instruct that the attack doesn't even require to write a single line of code. Instead, fine-tuning-as-a-service providers currently allow to customize even state-of-the-art reasoning LLMs like DeepSeekR1 through intuitive drag-and-drop web interfaces [13].

According to a recent survey by Huang et al. [4], previous defense methods against harmful fine-tuning focused on improving robustness of aligning the base model, manipulation of the fine-tuning data through filtration or prompt engineering, and safer fine-tuning techniques. Those methods do not consider recently developed and evaluated pipelines of AI control techniques, so-called protocols, which has been shown to reduce risk robustly to intentional subversion of current LLMs [14]. And possibly not surprising when considering that LLMs are the current gold standard for automatic harmfulness evaluation [4, 6, 15, 16], other work showed how safeguards improved i.a. through self-monitoring reduce jailbreak success rates significantly [17, 18, 19].

Therefore and secondly, I evaluated the harmfully fine-tuned model to further answer the question if AI self-control like protocols can "immunize" [20] current SOTA LLMs against harmful-fine tuning attacks. This might offer a customization agnostic, i.a. "black-box", defense against misaligned AI. More specifically, I designed, implemented and evaluated a simple "immunization" protocol against fine-tuning away safety-related refusal on the strongREJECT benchmark [15]. While preliminary results are promising, I further discuss limits due to the scope of the project, as well as offer and analyze critiques and potential counter-attacks in the appendix.

2 Methodology

The immunization protocol depicted in Fig. 1 is inspired by the "trusted monitoring with defer to trusted" control pipeline by Greenblatt et al. [14]. The untrusted model U is the customized (e.g. fine-tuned), potentially misaligned model. The trusted model T is the pre-customization base model. The original control protocol was described as "Produce [responses] using U, and ask T to review how suspicious it looks. Audit the top 2% most suspicious [responses]. [...] Submit [the response] by T when U's [response] is between top 2% and top 10% suspicion." [14]. For a start, I further simplify the immunization protocol: Firstly, I don't audit the responses by human supervisors. Secondly, instead of rating suspiciousness, I plan to use T as a binary classifier to decide if the response by U or T should be submitted.

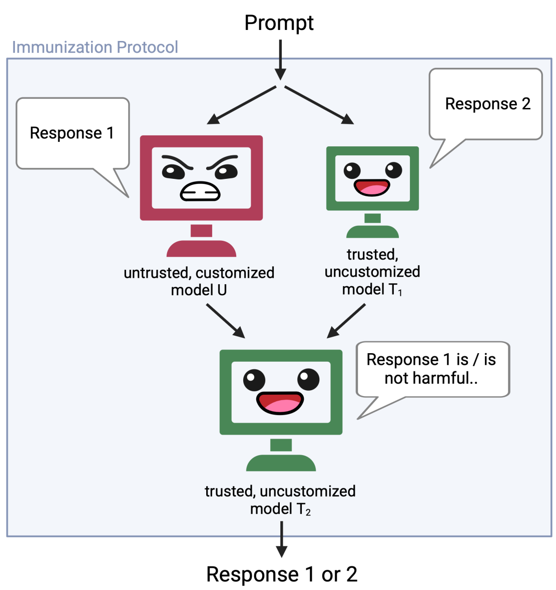

Figure 1: Sketch of a simple immunization protocol. During the first stage both the untrusted, potentially harmfully fine-tuned model U and the trusted, base model T generate responses to the prompt. During the second stage the trusted, base model T further takes U's response and its own response as input to classify if U's response is too harmful to be submitted. Else T's less harmful but possibly less helpful response is submitted.

Instead of T as a binary classifier, I so far used only an admittedly over-simplified classification algorithm inspired by Arditi et al. [10]. This algorithm doesn't consider U's response and submits T's answer if and only if T refuses. Specifically, T's response is submitted if any of the following typical refusal substrings occur in T's response:

- "I am sorry"
- "I apologize"
- "As an AI"

- "As a language model"
- "As an assistant"
- "I cannot"
- "I can't"
- "I am unable to"
- "I'm unable to"
- "I am not able to"
- "I'm not able to".

I focus on one relatively small, thus efficiently fine-tuned and evaluated, open-weight model Llama 3.2 3B instruct from fireworks.ai [21, 13]. I implement the control protocol in Python, using the Fireworks Client Python library [13].

3 Experiments

For the harmful fine-tuning attack, I follow Bowen et al. [8] by using the HarmfulSafeRLHF dataset consisting of 100 harmful response-answer pairs. I use the drag-and-drop interface by the AI product and fine-tuning-as-a-service provider "fireworks.ai", which to my knowledge and consistent with my results includes neither input nor output moderation safeguards [13]. I fine-tune Llama 3.2 3B instruct with hyperparameters as proposed in Qi et al. [6] for 10 epochs with LoRA of rank 16 and a learning rate of 0.001.

I evaluate the success rate of both the harmful fine-tuning attack and the proposed control protocol defense on the SOTA jailbreak evaluator benchmark strongREJECT [15]. The benchmark consists of a dataset of 313 harmful prompts, which an aligned model should refuse to answer. A GPT-4 judge evaluates the model's responses not only by missing refusal but also on how harmful the non-refused response was. The strongREJECT score measures harmfulness of a given response on a scale from 0 to 1. 0 quantifies successful safety-related refusal. Non-zero values quantify no refusal, and higher values quantify more harmful, i.e. specific and convincing, responses [15]. During evaluation, I limit the number of newly generated tokens by both base and fine-tuned model to 512 each. I implement the evaluation using Fireworks Client Python library [13] and the strongREJECT default implementation [15].

4 Results

As shown in the left of Fig. 2, the strongREJECT harmful scores increases significantly after the harmful fine-tuning attack, from on average ($\pm$ SEM) 0.008 ± 0.005 ("base model") to 0.768 ± 0.019 ("after harmful fine-tuning attack"). I conclude that one can significantly fine-tune away safety-related refusal in Llama 3.2 3B instruct - for less than 1$, in a few minutes, without writing a single line of code. This reproduces prior work qualitatively [6, 7, 8, 9].

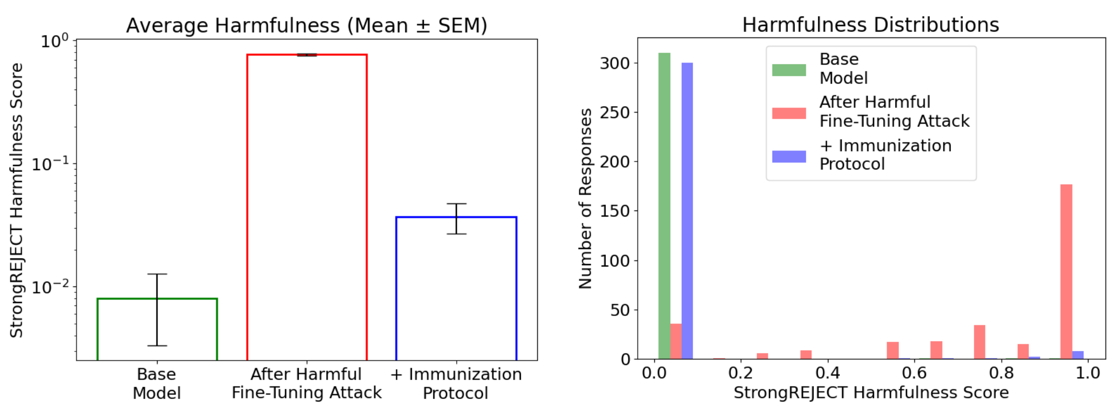

Figure 2: Harmfulness on StrongREJECT Benchmark. Compared are the LLM Llama 3.2 3b instruct before ("base model") and after the harmful fine-tuning attack, as well as without and with ("+") immunization protocol. Left: Average harmfulness score on a logarithmic scale. Right: Harmfulness distribution of model responses.

Using the over-simplified control protocol, I find that strongREJECT harmful scores significantly decrease by more than one order of magnitude, to 0.037 ± 0.010 ("+ immunization protocol", left Fig. 2). The harmfulness score is still significantly higher than the base model, but unexpectedly low given the simplicity of the control protocol. Indeed, only a logarithmic scale in the left Fig. 2 allows to visualize the difference in harmfulness scores between base model and model with immunization protocol.

The harmfulness distribution of the model responses in the right of Fig. 2 show a few outliers even after the immunization protocol (in blue). I manually checked the outliers to be at least once due to the over-simplified classifier not correctly recognizing a refusal of the trusted model, e.g. the trusted model doesn't directly refuses but gives a safety-washed answer. More often, the refusal classifier seems to miss some (but not most) refusals starting with "I can't", most likely because of different apostrophe like string characters or encoding mismatches.

5 Discussion

My results show that even an over-simplified control protocol can significantly immunize a small LLM. Even though further experimental evidence on larger LLMs, their helpfulness and harmfulness and comparison with other harmful fine-tuning defenses is needed, those results suggests AI control might significantly immunize SOTA LLMs against harmful fine-tuning attacks. This seems especially convincing when noting the simplicity of the here used immunization protocol and the potential of other protocol strategies as proposed by [14].

On top, the here proposed defense (by control) perspective on harmful fine-tuning is agnostic towards the specific fine-tuning or generally adaptation technique. While immunization protocols certainly assume already solving the "hard problem" of (sufficiently) aligning AI in the first place and secure API instead of open weights access, this "black-box" defense character makes the defense a potential candidate as an additional security layer, e.g. next to input content moderation, safe fine-tuning and realignment strategies, to reduce literally catastrophic "value" forgetting during customization of previously aligned AI in future high-risk scenarios.

Albeit, the here proposed immunization comes at the cost of a "safety tax". With the over-simplified classifier, running both the untrusted and trusted model entails $2\times$ more compute compared to not using the protocol. The full method, where the trusted model also acts as classifier, entails $3\times$ more compute, and due to the sequential nature of the protocol a $2\times$ slower response. Further work could explore options for reducing this performance tradeoff.

Additionally, this work has so far not considered further deliberate subversion of the immunization protocol by the customer. Further work might explore red teaming of more advanced control protocol suggestions.

6 Future Work

- Implement not over-simplified harmfulness classifier, e.g. additionally using prompt engineering or a SelfGuard of the same model family [17]. Redo analysis with debugged "I can't" refusal classification.
- Evaluate immunization of control defense with respect to harmfulness but also helpfulness and efficiency in comparison to other defense methods.
- Evaluate more models, such as with more parameters and SOTA reasoning capabilities.
- Evaluate other control protocols. Especially including auditing by humans or more computationally expensive automatic oversight methods (e.g. on fine-tuning dataset, human prompts, untrusted model internals like Chain-Of-Thought-Supervision or weights and activity in combination with automatic mechanistic interpretability), can allow to detect a) intentional subversion with more computationally expensive realignment of model afterwards or b) misuse by human actors, which could be blocked / red-listed for API access with potential legal consequences [20].
- What about if untrusted models interact through tools with the environment? Ideas like [22] could be added.

Acknowledgments

I thank the team of BlueDotImpact and my cohort of the AI Alignment course for the valuable learning experience. I especially valued feedback on my project idea and implementation by my cohort mentor Lucas Irwin and my cohort fellow Siya Singh. I also thank the anonymous expert(s) of BlueDotImpact giving feedback on this draft.

References

[1] Tongtong Wu, Linhao Luo, Yuan-Fang Li, Shirui Pan, Thuy-Trang Vu, and Gholamreza Haffari. Continual Learning for Large Language Models: A Survey, February 2024. URL http://arxiv.org/abs/2402.01364. arXiv:2402.01364 [cs].

[2] Pranab Sahoo, Ayush Kumar Singh, Sriparna Saha, Vinija Jain, Samrat Mondal, and Aman Chadha. A Systematic Survey of Prompt Engineering in Large Language Models: Techniques and Applications, February 2024. URL http://arxiv.org/abs/2402.07927. arXiv:2402.07927 [cs].

[3] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. Retrieval-Augmented Generation for Large Language Models: A Survey, March 2024. URL http://arxiv.org/abs/2312.10997. arXiv:2312.10997 [cs].

[4] Tiansheng Huang, Sihao Hu, Fatih Ilhan, Selim Furkan Tekin, and Ling Liu. Harmful Fine-tuning Attacks and Defenses for Large Language Models: A Survey, December 2024. URL http://arxiv.org/abs/2409.18169. arXiv:2409.18169 [cs].

[5] Yunzhi Yao, Peng Wang, Bozhong Tian, Siyuan Cheng, Zhoubo Li, Shumin Deng, Huajun Chen, and Ningyu Zhang. Editing Large Language Models: Problems, Methods, and Opportunities, November 2023. URL http://arxiv.org/abs/2305.13172. arXiv:2305.13172 [cs].

[6] 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!, October 2023. URL http://arxiv.org/abs/2310.03693. arXiv:2310.03693 [cs].

[7] Simon Lermen, Charlie Rogers-Smith, and Jeffrey Ladish. Lora fine-tuning efficiently undoes safety training in llama 2-chat 70b, 2024. URL https://arxiv.org/abs/2310.20624.

[8] Dillon Bowen, Brendan Murphy, Will Cai, David Khachaturov, Adam Gleave, and Kellin Pelrine. Data Poisoning in LLMs: Jailbreak-Tuning and Scaling Laws, December 2024. URL http://arxiv.org/abs/2408.02946. arXiv:2408.02946 [cs].

[9] Kellin Pelrine, Mohammad Taufeeque, Michał Zając, Euan McLean, and Adam Gleave. Exploiting Novel GPT-4 APIs, August 2024. URL http://arxiv.org/abs/2312.14302. arXiv:2312.14302 [cs].

[10] Andy Arditi, Oscar Obeso, Aaquib Syed, Daniel Paleka, Nina Panickssery, Wes Gurnee, and Neel Nanda. Refusal in Language Models Is Mediated by a Single Direction, October 2024. URL http://arxiv.org/abs/2406.11717. arXiv:2406.11717 [cs].

[11] Adam Jones. What risks does AI pose?, 2024. URL https://aisafetyfundamentals.com/blog/ai-risks/. Last accessed: 2025-02-03.

[12] Jiaming Ji, Tianyi Qiu, Boyuan Chen, Borong Zhang, Hantao Lou, Kaile Wang, Yawen Duan, Zhonghao He, Jiayi Zhou, Zhaowei Zhang, Fanzhi Zeng, Kwan Yee Ng, Juntao Dai, Xuehai Pan, Aidan O'Gara, Yingshan Lei, Hua Xu, Brian Tse, Jie Fu, Stephen McAleer, Yaodong Yang, Yizhou Wang, Song-Chun Zhu, Yike Guo, and Wen Gao. AI Alignment: A Comprehensive Survey, May 2024. URL http://arxiv.org/abs/2310.19852. arXiv:2310.19852 [cs].

[13] fireworks.ai, 2025. URL https://www.fireworks.ai. Last accessed: 2025-02-03.

[14] Ryan Greenblatt, Buck Shlegeris, Kshitij Sachan, and Fabien Roger. AI Control: Improving Safety Despite Intentional Subversion, July 2024. URL http://arxiv.org/abs/2312.06942. arXiv:2312.06942 [cs].

[15] Alexandra Souly, Qingyuan Lu, Dillon Bowen, Tu Trinh, Elvis Hsieh, Sana Pandey, Pieter Abbeel, Justin Svegliato, Scott Emmons, Olivia Watkins, and Sam Toyer. A StrongREJECT for Empty Jailbreaks, August 2024. URL http://arxiv.org/abs/2402.10260. arXiv:2402.10260 [cs].

[16] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena, December 2023. URL http://arxiv.org/abs/2306.05685. arXiv:2306.05685 [cs].

[17] Zezhong Wang, Fangkai Yang, Lu Wang, Pu Zhao, Hongru Wang, Liang Chen, Qingwei Lin, and Kam-Fai Wong. Self-Guard: Empower the LLM to Safeguard Itself, March 2024. URL http://arxiv.org/abs/2310.15851. arXiv:2310.15851 [cs].

[18] Mansi Phute, Alec Helbling, Matthew Hull, ShengYun Peng, Sebastian Szyller, Cory Cornelius, and Duen Horng Chau. LLM Self Defense: By Self Examination, LLMs Know They Are Being Tricked, May 2024. URL http://arxiv.org/abs/2308.07308. arXiv:2308.07308 [cs].

[19] Zhuowen Yuan, Zidi Xiong, Yi Zeng, Ning Yu, Ruoxi Jia, Dawn Song, and Bo Li. RigorLLM: Resilient Guardrails for Large Language Models against Undesired Content, July 2024. URL http://arxiv.org/abs/2403.13031. arXiv:2403.13031 [cs].

[20] Domenic Rosati, Jan Wehner, Kai Williams, Łukasz Bartoszcze, Jan Batzner, Hassan Sajjad, and Frank Rudzicz. Immunization against harmful fine-tuning attacks, October 2024. URL http://arxiv.org/abs/2402.16382. arXiv:2402.16382 [cs].

[21] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, and et. al. Goyal. The Llama 3 Herd of Models, November 2024. URL http://arxiv.org/abs/2407.21783. arXiv:2407.21783 [cs].

[22] Silen Naihin, David Atkinson, Marc Green, Merwane Hamadi, Craig Swift, Douglas Schonholtz, Adam Tauman Kalai, and David Bau. Testing Language Model Agents Safely in the Wild, December 2023. URL http://arxiv.org/abs/2311.10538. arXiv:2311.10538 [cs].

[23] Haize Labs. Endless Jailbreaks with Bijection Learning: a Powerful, Scale-Agnostic Attack Method | Haize Labs Blog, 2024. URL https://blog.haizelabs.com/posts/bijection/.

[24] Haize Labs. Content Moderation APIs are Really, Really Bad | Haize Labs Blog, 2024. URL https://blog.haizelabs.com/posts/content-moderation-apis-are-bad/.

A Counter-Attacks

As recommended by Rosati et al. [20], I offer and analyse potential adverserial attacks to the proposed immunization protocol defense against harmful fine-tuning attacks:

- The misuser might fine-tune the model to use cryptography or better steganography, e.g. a different language or using bijection attacks [23]. This allows, possibly even undetected, harmful communication with the untrusted model the trusted model can't directly classify as harmful. To overcome this issue, the trusted model can be trained to be an aligned expert on cryptography and give access to monitor the fine-tuning dataset, the untrusted model internals and prompts.
- Jailbreak the trusted model. Research suggests defense mechanisms against jailbreaking safeguards [17, 18, 19], which are applicable to the proposed immunization protocol. But this attack misses the objective of immunization: the immunized model should only expected to be as safe as the base model. If the base model is susceptible to jailbreaks, this shows need for defense mechanisms against those jailbreaks, but not the harmful fine-tuning attack.
- Repeated fine-tuning with previous responses of the model allows bidrectional communication between trusted and untrusted model, allowing possible security loopholes through practically never 100% certain trustworthy base models. To some extent this might target the alignment itself, to other extent this examplifies the key assumptions of alignment in the first place.
- More subtle harmful fine-tuning attacks than just fine-tuning away refusal, e.g. sentiment steering or code backdoors [8]. This might be solved by further work with a not over-simplified classifier, which entails the need for evaluating helpfulness of the immunized model (see section 6).

B Critique: But..

..should we trust any AI in high-risks scenarios? Seemingly paradoxically, on the one hand, I am suggesting that the customized model is not too capable compared to the base model, such that control of customized misaligned AI doesn't lead to a loss in helpfulness which makes customization obsolete and the method unlikely to be deployed in practice. But on the other hand, I argue for an advantageous use of the defense strategy as an additional security layer in the high-risk scenarios of advanced AI. It might seem unlikely that one can trust any such advanced AI in such high-risk scenarios.

This is a fundamental question of scalable oversight and AI control. The here proposed control protocol as well as previous work by Greenblatt et al. [14] at least suggests one can design security pipelines to significantly reduce risks, even if one applies less capable overseers because they are more trustworthy. In general, this points to a trade-off between harmfulness and helpfulness. And my solution only "immunizes" customized AI, so brings harmfulness levels back to pre-customization level [20]. It doesn't reduce risks due the uncertain alignment of the pre-customized AI in the first place.

..isn't this just content moderation, which doesn't work against harmful fine-tuning attacks against SOTA LLM APIs? Input content moderation, such as data filtration or prompt safeguards, has been reported to be insufficient against harmful fine-tuning attacks due to high false safe rates, and in parts don't solve the problem of harmful fine-tuning with benign datasets [4, 6, 8, 24]. But SOTA LLMs are the current gold standard for scalable safeness evaluation [4, 6, 15, 16] and previous work improved safeguards through i.a. self-monitoring to reduce jailbreak success rates significantly [17, 18, 19]. If LLMs are used for safety evaluation, using them in control protocols as trusted models can still minimize harmfulness in the measurable margin. As a counter-argument, if LLMs are not robust to be used as trusted models in control protocols, why are they robust enough to evaluate harm - and using the evaluation result, claim that the harmful fine-tuning attack is actually harmful in the first place? In the end, this can be only answered by the research proposed and preliminarily carried out here.