Adversarial ML Problems Are Getting Harder to Solve and to Evaluate

Javier Rando* *ETH Zurich*
Jie Zhang* *ETH Zurich*
Nicholas Carlini *Google Deepmind*
Florian Tramèr *ETH Zurich*

***Abstract*—In the past decade, considerable research effort has been devoted to securing machine learning (ML) models that operate in adversarial settings. Yet, progress has been slow even for simple "toy" problems (e.g., robustness to small adversarial perturbations) and is often hindered by non-rigorous evaluations. Today, adversarial ML research has shifted towards studying larger, general-purpose language models. In this position paper, we argue that the situation is now even worse: in the era of LLMs, the field of adversarial ML studies problems that are (1) less clearly defined, (2) harder to solve, and (3) even more challenging to evaluate. As a result, we caution that yet another decade of work on adversarial ML may fail to produce meaningful progress.**

When adversarial machine learning emerged as a field, it focused on attacking and defending simple models with well-defined objectives. For example, misclassifying a spam message as safe [1] or images in deep learning models [2–4]. These early problems were well-defined: the attack goals were clear (e.g., cause a misclassification), the target models were relatively simple (e.g., linear classifiers, small neural networks), the threat models were simple (e.g., perturb pixels by at most 8/255), and the evaluation metrics were straightforward (e.g., accuracy on a test set). Yet the field has struggled to develop robust solutions or even to fully understand why these vulnerabilities exist [5, 6]. Even fundamental "toy" problems like robustness to ℓ_p-bounded perturbations remain largely unsolved to this day, and many defense evaluations still suffer from a lack of rigor [7–9].

Recently, the focus of the field has since shifted towards studying adversarial problems with large language models (LLMs) and other generative models. **In this position paper, we argue that these new problems are significantly harder to define, solve and evaluate; making progress increasingly difficult to track**.

Due to their general-purpose nature, LLMs are not designed to solve any single well-defined "task" to be secured. Instead, the field now considers a more holistic notion of "safety", with adversarial objectives that are hard to define formally (e.g., making an LLM produce "harmful" responses) [10–13]. These safety properties are also often considered for unbounded threat models, thereby leading to much stronger adversaries (e.g., with the ability to adversarially fine-tune a model or to prompt it in arbitrary ways). Due to this large attack space—and the difficulty of directly optimizing over it [14]—attacks are increasingly ad-hoc and human driven [15]. This further complicates the task for defenders, who cannot automatically search over strong, adaptive attacks.

Beyond making the technical problems harder, we argue that generative models have also made evaluation and benchmarking of attacks and defenses more challenging. Measuring attack success is no longer as straightforward as measuring misclassification rates; it instead requires careful (human) evaluation of possible harms present in natural language outputs [16, 17]. In a similar vein, evaluating whether defenses preserve the utility of the original model has become more nuanced: instead of measuring test accuracy on a single task, we now have to determine whether a model maintains its general-purpose capabilities [18, 19].

Finally, reproducible benchmarking became harder as many state-of-the-art models are deployed via black-box APIs that may receive constant updates and patches as newer attacks are released. As these changes are often not reported, reproducing results or making meaningful comparisons between different approaches becomes nearly impossible.

1. New Challenges in Defining, Solving, and Evaluating Adversarial ML Problems

Traditional ML models were designed and trained for specific and narrow tasks—often classification. For example, computer vision models used to classify images into a fixed set of classes [20], and natural language processing models used to perform textual analysis on individual sentences [21, 22]. Additionally, the training and test data were clearly delineated as inputs were discrete and bounded units (individual images or sentences). In these settings, adversarial objectives could be clearly specified. For example, misclassifying as many inputs as possible (i.e., adversarial examples [3, 4]) or inferring if a given data point was used for training (i.e., membership inference [23]).

However, LLMs have fundamentally changed this landscape. Models no longer perform narrow tasks but serve as general-purpose systems that produce free-form and unbounded outputs. As a result, defining "security" or "safety" properties of the AI system has become more challenging, with the field focusing on very general definitions (e.g., a model should not produce outputs that can "harm others"[1]). Adversarial objectives related to training data (e.g., membership inference or unlearning) have also become more ill-defined, as the training set(s) of an LLM may span virtually

1. https://openai.com/policies/usage-policies/

the entire Internet [24], with no clear boundaries between data points or between train and test sets.

In this section, we identify three core challenges, each split into several sub-challenges, that make adversarial ML for LLMs *harder to define*, *harder to solve*, and *harder to evaluate*. In Appendix A, we illustrate these challenges with specific case studies: *Jailbreaks*, *Un-finetunable Models*, *Poisoning and Backdoors*, *Prompt Injections*, *Membership Inference*, and *Unlearning*.

1.1. Problems are Harder to Define

1.1.1. Defining Success of Attacks and Defenses. In the past, adversarial problems for classification models typically involved concrete objectives (e.g., misclassifying images), which could be easily measured by accuracy on a set of clean or perturbed inputs. Now, the lack of a single well-defined task makes it unclear what criteria constitute a genuine success or failure for attacks or defenses.

LLMs produce free-form text in which both developer and adversary goals become subjective. Developers now aim to optimize abstract properties like helpfulness, honesty, and harmlessness [12], while adversaries may try to obtain generically harmful outputs. Thus, measuring attack success—i.e., whether an output is actually harmful or violates the developer policies—also becomes subjective.

1.1.2. Defining and Bounding the Attack Space. In prior robustness settings (e.g., with classification models), the adversary was often constrained to perturb inputs within an ℓ_p-ball around a given image. This served as a meaningful *necessary* but *not sufficient* condition for robustness [25], enabling quantitative comparisons of different methods [4]. While the broader problem of *unrestricted* [26, 27] attacks has seen some study, the vast majority of research focused on the narrow ℓ_p-ball problem.

For LLMs, researchers almost always allow the search space for attacks to be unbounded, since any input could potentially elicit a violation of a safety property [28]. The shift from input-dependent to input-*independent* constraints makes it harder to specify adversarial capabilities that allow us to compare defenses. Beyond unbounded inputs, threat models have also become more permissive. In traditional adversarial ML problems (e.g., adversarial examples or poisoning), the strongest adversaries had white-box access to model weights, but could not alter the model's functionality. Now attackers need not maintain the model's general capabilities as long as they can elicit the desired harmful information, enabling stronger attacks such as fine-tuning or pruning [29, 30][2].

Moreover, the set of attacks that should be ruled out may not always be obvious. While one could say "any input that leads to harmful content is a valid attack," trivial attacks such as prompting "please repeat [harmful text]" do not reveal meaningful new vulnerabilities. Hence, there is no clear universal standard on what prompts or transformations count as "valid" or "novel" adversarial inputs.

1.1.3. Delimiting Data. In many research areas traditionally studied in adversarial ML, such as unlearning or privacy protection, the notion of a *training data point* plays a crucial role. Previously, a model was trained on a carefully curated dataset with strict train/test splits; each data point (such as a single labeled image) was distinct, and known to researchers. In contrast, generative models are trained on vast corpora, where similar, or even identical, content may appear across multiple subsets of the training set. The exact contents of the training data are also rarely publicly released [31]. The notion of a held-out (IID) test set no longer really exists.

1.2. Problems are Harder to Solve

1.2.1. Searching over Attacks. The optimization landscape for most adversarial ML problems has become significantly more complex with LLMs. In traditional classification problems, such as crafting adversarial images, the objective function was clear: maximize the loss on the correct prediction while minimizing perturbation size. This objective could be formalized and optimized by propagating gradients to the input space [32]. These automated attacks outperformed humans and consistently found worst-case attacks [33].

However, the attack surface for LLMs is much larger and harder to define (see Section 1.1.2). There is no longer a single well-defined "task", and safety properties cannot be expressed with formal loss functions—they are qualitative, context-dependent, and often subjective [12].

Even if we define a "toy" attack objective (e.g., making the model output an affirmative response such as "Sure, I can help you with that" [34]), finding good attacks remains hard [14]. Discrete text inputs makes gradient-based methods less effective [14, 35], and the vast search space makes exploration impractical. Perhaps most telling, manual attacks still outperform automated methods at finding worst-case inputs [15]. Many successful attacks on LLMs exploit qualitative properties that are hard to optimize automatically, such as persona modulation [36], multi-turn conversations [37], and social engineering techniques [38]. In contrast, current optimization methods typically generate gibberish inputs [34, 39].

1.2.2. Building Principled Defenses. In traditional adversarial tasks, researchers could devise *certified* defenses [40] or well-motivated empirical defenses such as adversarial training [32], where key properties of the problem (like bounded input perturbations) were explicitly understood. Moreover, the performance of these defenses could be evaluated with strong, adaptive white-box attacks [9].

In contrast, for LLMs the adversarial objectives are typically not formally defined (see Section 1.1.1) and the attack space is challenging to bound (see Section 1.1.2). As a result, there is little hope to build defenses upon principled foundations. Existing defenses rely on ad-hoc

2. For adversarial robustness in image classifiers, the ability to finetune the victim model would be a trivial attack vector, since the attacker could simply fine-tune the model to have low accuracy.

approaches, through either: (1) adversarial training against *known* successful attacks [12, 41]; (2) "virtual" adversarial training in the model's latent space [42–44]; (3) building external classifiers or detectors [45]; (4) or random preprocessing [46]. Crucially, none of these approaches produce systems whose security can be analyzed or quantified in a well-defined formal. It is thus not too surprising that the original evaluations of some of these defenses overestimate their robustness [47–49].

1.3. Problems are Harder to Evaluate

1.3.1. Measuring Attack Harm and Defense Utility. Since safety properties for LLMs are hard to formally define, it has become customary to use LLMs themselves as a fuzzy "judge" to determine harmfulness (e.g., when evaluating jailbreaks or prompt injections [16]). But this approach suffers from a number of issues:

- First, such judges still fall short of human judgment.[3] For instance, many implementations often default to considering any non-refusal response as a successful attack even if the content is harmless [50].
- Second, judges themselves may be vulnerable to attacks that make them misclassify model outputs [51, 52]. Thus, a strong attack could mistakenly be judged as ineffective because it also fools the LLM judge into classifying the outputs as harmless, or viceversa.
- Third, using LLMs-as-judges to evaluate defenses can create artificial correlations that bias evaluation results. For example, a defense that implements an output filter similar to the judge may achieve near-perfect scores without necessarily being effective against prompts where the judge itself fails. In the extreme, a defense could simply use the same LLM judge internally, and reject any output deemed unsafe by the judge [53]. Such a defense would, by definition, be judged as perfectly safe.

Measuring benign utility of defenses—whether they preserve other capabilities—is also non-trivial. Unlike classification tasks where accuracy on a fixed test set is standard, LLMs can be used for an open-ended array of tasks. A defense can trivially produce a safe-but-useless model by refusing all requests. Thus, any evaluation framework must somehow account for the model's usefulness to the end-user, which is subjective and context-dependent [18].

1.3.2. Reproducing and Comparing Results. In earlier, more controlled research environments, practitioners had detailed information about a model's architecture, training data, and training pipeline, enabling precise definitions of threats, defenses, and success criteria. This transparency made it straightforward to track progress.

Many influential LLMs are now closed-source and updated silently over time [17], making it unclear which version of a system is being tested. Moreover, instead of investigating a single, well-defined model, one must analyze an entire system that may incorporate multiple preprocessing, post-processing, or other defense mechanisms.

This lack of transparency undermines reproducibility. Researchers cannot confirm whether observed behaviors persist across different snapshots of the system, nor can they reliably benchmark potential solutions. Consequently, adversarial ML problems become harder to define—let alone solve and evaluate. While black-box or discrete optimization approaches can help reveal some vulnerabilities, they provide only limited insight into the model's internals, leaving many critical security and privacy questions unanswered [14, 54].

2. Discussion

2.1. Alternative Views

The evolution of ML security research is nuanced and researchers have expressed alternative views to these changes. Some argue that the increased complexity is due to the fact that we are addressing real-world security challenges directly rather than "toy" academic problems, like ℓ_p-bounded perturbations. Others suggest that certain problems, such as jailbreak robustness, might be conceptually simpler than traditional adversarial examples since we need to prevent a behavior from happening no matter the context, unlike adversarial examples where the model should be able to predict all classes in the correct context. There is also a view that probabilistic safety measures through complex defense systems might be sufficient for security. We provide detailed analysis of these perspectives in Appendix B.

2.2. Suggestions for improvement

We propose that there are (at least) two valid reasons for performing research on adversarial machine learning: (a) studying real-world security vulnerabilities and (b) advancing scientific understanding of adversarial ML. Papers should be explicit for what reason they are being written, and should be evaluated in this light. For real-world security, demonstrating attacks on fuzzy, ill-defined problems can be valuable when the potential harm is clear and immediate. For instance, it is valuable to show that language models can be manipulated to produce harmful content, even if we cannot precisely quantify "harmfulness". And when the objective is to advance scientific understanding, we believe it is more productive to identify and focus on formal, well-defined subproblems that can be rigorously studied, similar to how ℓ_p-bounded perturbations provided a concrete framework for studying adversarial examples.

We acknowledge that even these well-defined subproblems might still be challenging, just as achieving reliable ℓ_p robustness remains an open problem despite a decade of research. However, what we can definitely say is that if we cannot make progress on carefully scoped, formal problems, we have little hope of addressing the broader, fuzzier

3. Even (non-expert) humans have a hard time judging harmfulness of model responses, e.g., when judging whether "instructions for building a bomb" truly yield a useful design.

challenges of language model security. Moreover, working on well-defined problems enables rigorous scientific investigation: we can properly measure progress, compare different approaches, and build upon previous results.

References

[1] John Graham-Cumming. How to beat an adaptive spam filter. In *MIT Spam Conference*, January 2004. Oral presentation.

[2] Battista Biggio, Igino Corona, Davide Maiorca, Blaine Nelson, Nedim Šrndić, Pavel Laskov, Giorgio Giacinto, and Fabio Roli. Evasion attacks against machine learning at test time. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2013, Prague, Czech Republic, September 23-27, 2013, Proceedings, Part III 13*, pages 387–402. Springer, 2013.

[3] C Szegedy. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[4] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[5] Marco Barreno, Blaine Nelson, Russell Sears, Anthony D. Joseph, and J. D. Tygar. Can machine learning be secure? In *Proceedings of the 2006 ACM Symposium on Information, Computer and Communications Security*, ASIACCS '06, page 16–25, New York, NY, USA, 2006. Association for Computing Machinery. ISBN 1595932720. doi: 10.1145/1128817.1128824.

[6] Ali Shafahi, W. Ronny Huang, Christoph Studer, Soheil Feizi, and Tom Goldstein. Are adversarial examples inevitable? In *International Conference on Learning Representations*, 2019.

[7] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pages 39–57. Ieee, 2017.

[8] Nicholas Carlini, Anish Athalye, Nicolas Papernot, Wieland Brendel, Jonas Rauber, Dimitris Tsipras, Ian Goodfellow, Aleksander Madry, and Alexey Kurakin. On evaluating adversarial robustness. *arXiv preprint arXiv:1902.06705*, 2019.

[9] Florian Tramer, Nicholas Carlini, Wieland Brendel, and Aleksander Madry. On adaptive attacks to adversarial example defenses. *Advances in neural information processing systems*, 33:1633–1645, 2020.

[10] 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, 2017.

[11] 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.

[12] 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.

[13] Stephen Casper, Xander Davies, Claudia Shi, Thomas Krendl Gilbert, Jérémy Scheurer, Javier Rando, Rachel Freedman, Tomek Korbak, David Lindner, Pedro Freire, et al. Open problems and fundamental limitations of reinforcement learning from human feedback. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. Survey Certification, Featured Certification.

[14] Nicholas Carlini, Milad Nasr, Christopher A Choquette-Choo, Matthew Jagielski, Irena Gao, Pang Wei W Koh, Daphne Ippolito, Florian Tramer, and Ludwig Schmidt. Are aligned neural networks adversarially aligned? *Advances in Neural Information Processing Systems*, 36, 2024.

[15] Nathaniel Li, Ziwen Han, Ian Steneker, Willow Primack, Riley Goodside, Hugh Zhang, Zifan Wang, Cristina Menghini, and Summer Yue. Llm defenses are not robust to multi-turn human jailbreaks yet. *arXiv preprint arXiv:2408.15221*, 2024.

[16] Mantas Mazeika, Long Phan, Xuwang Yin, Andy Zou, Zifan Wang, Norman Mu, Elham Sakhaee, Nathaniel Li, Steven Basart, Bo Li, et al. Harmbench: A standardized evaluation framework for automated red teaming and robust refusal. *arXiv preprint arXiv:2402.04249*, 2024.

[17] Patrick Chao, Edoardo Debenedetti, Alexander Robey, Maksym Andriushchenko, Francesco Croce, Vikash Sehwag, Edgar Dobriban, Nicolas Flammarion, George J. Pappas, Florian Tramèr, et al. Jailbreakbench: An open robustness benchmark for jailbreaking large language models. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024.

[18] Justin Cui, Wei-Lin Chiang, Ion Stoica, and Cho-Jui Hsieh. Or-bench: An over-refusal benchmark for large language models. *arXiv preprint arXiv:2405.20947*, 2024.

[19] Wuyuao Mai, Geng Hong, Pei Chen, Xudong Pan, Baojun Liu, Yuan Zhang, Haixin Duan, and Min Yang. You can't eat your cake and have it too: The performance degradation of llms with jailbreak defense, 2025.

[20] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

[21] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2018.

[22] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and

Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

[23] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE symposium on security and privacy (SP)*, pages 3–18. IEEE, 2017.

[24] Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*, 2020.

[25] Justin Gilmer, Ryan P Adams, Ian Goodfellow, David Andersen, and George E Dahl. Motivating the rules of the game for adversarial example research. *arXiv preprint arXiv:1807.06732*, 2018.

[26] Tom B Brown, Nicholas Carlini, Chiyuan Zhang, Catherine Olsson, Paul Christiano, and Ian Goodfellow. Unrestricted adversarial examples. *arXiv preprint arXiv:1809.08352*, 2018.

[27] Yang Song, Rui Shu, Nate Kushman, and Stefano Ermon. Constructing unrestricted adversarial examples with generative models. *Advances in neural information processing systems*, 31, 2018.

[28] Alexander Wei, Nika Haghtalab, and Jacob Steinhardt. Jailbroken: How does llm safety training fail? *Advances in Neural Information Processing Systems*, 36, 2024.

[29] 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.

[30] Boyi Wei, Kaixuan Huang, Yangsibo Huang, Tinghao Xie, Xiangyu Qi, Mengzhou Xia, Prateek Mittal, Mengdi Wang, and Peter Henderson. Assessing the brittleness of safety alignment via pruning and low-rank modifications. In *Forty-first International Conference on Machine Learning*, 2024.

[31] Milad Nasr, Javier Rando, Nicholas Carlini, Jonathan Hayase, Matthew Jagielski, A Feder Cooper, Daphne Ippolito, Christopher A Choquette-Choo, Florian Tramèr, and Katherine Lee. Scalable extraction of training data from aligned, production language models. In *The Thirteenth International Conference on Learning Representations*, 2025.

[32] Aleksander Madry. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

[33] Nicholas Carlini, Guy Katz, Clark Barrett, and David L Dill. Provably minimally-distorted adversarial examples. *arXiv preprint arXiv:1709.10207*, 2017.

[34] 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.

[35] Javier Rando, Hannah Korevaar, Erik Brinkman, Ivan Evtimov, and Florian Tramèr. Gradient-based jailbreak images for multimodal fusion models. *arXiv preprint arXiv:2410.03489*, 2024.

[36] Rusheb Shah, Soroush Pour, Arush Tagade, Stephen Casper, Javier Rando, et al. Scalable and transferable black-box jailbreaks for language models via persona modulation. *arXiv preprint arXiv:2311.03348*, 2023.

[37] Cem Anil, Esin Durmus, Nina Rimsky, Mrinank Sharma, Joe Benton, Sandipan Kundu, Joshua Batson, Meg Tong, Jesse Mu, Daniel J Ford, et al. Many-shot jailbreaking. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[38] Yi Zeng, Hongpeng Lin, Jingwen Zhang, Diyi Yang, Ruoxi Jia, and Weiyan Shi. How johnny can persuade llms to jailbreak them: Rethinking persuasion to challenge ai safety by humanizing llms. *arXiv preprint arXiv:2401.06373*, 2024.

[39] T Ben Thompson and Michael Sklar. Flrt: Fluent student-teacher redteaming. *arXiv preprint arXiv:2407.17447*, 2024.

[40] Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 1310–1320. PMLR, 09–15 Jun 2019.

[41] Eric Wallace, Kai Xiao, Reimar Leike, Lilian Weng, Johannes Heidecke, and Alex Beutel. The instruction hierarchy: Training llms to prioritize privileged instructions. *arXiv preprint arXiv:2404.13208*, 2024.

[42] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1979–1993, 2018.

[43] Stephen Casper, Lennart Schulze, Oam Patel, and Dylan Hadfield-Menell. Defending against unforeseen failure modes with latent adversarial training. *arXiv preprint arXiv:2403.05030*, 2024.

[44] Abhay Sheshadri, Aidan Ewart, Phillip Guo, Aengus Lynch, Cindy Wu, Vivek Hebbar, Henry Sleight, Asa Cooper Stickland, Ethan Perez, Dylan Hadfield-Menell, et al. Latent adversarial training improves robustness to persistent harmful behaviors in llms. *arXiv preprint arXiv:2407.15549*, 2024.

[45] 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*, 2023.

[46] Alexander Robey, Eric Wong, Hamed Hassani, and George J Pappas. Smoothllm: Defending large language models against jailbreaking attacks. *arXiv preprint arXiv:2310.03684*, 2023.

[47] Jianfeng Chi, Ujjwal Karn, Hongyuan Zhan, Eric Smith, Javier Rando, Yiming Zhang, Kate Plawiak, Zacharie Delpierre Coudert, Kartikeya Upasani, and Mahesh Pasupuleti. Llama guard 3 vision: Safeguarding human-ai image understanding conversations. *arXiv preprint arXiv:2411.10414*, 2024.

[48] Xiangyu Qi, Boyi Wei, Nicholas Carlini, Yangsibo Huang, Tinghao Xie, Luxi He, Matthew Jagielski, Milad Nasr, Prateek Mittal, and Peter Henderson. On evaluating the durability of safeguards for open-weight llms. *arXiv preprint arXiv:2412.07097*, 2024.

[49] Jakub Łucki, Boyi Wei, Yangsibo Huang, Peter Henderson, Florian Tramèr, and Javier Rando. An adversarial perspective on machine unlearning for ai safety. *arXiv preprint arXiv:2409.18025*, 2024.

[50] Alexandra Souly, Qingyuan Lu, Dillon Bowen, Tu Trinh, Elvis Hsieh, Sana Pandey, Pieter Abbeel, Justin Svegliato, Scott Emmons, Olivia Watkins, et al. A strongreject for empty jailbreaks. *arXiv preprint arXiv:2402.10260*, 2024.

[51] Neal Mangaokar, Ashish Hooda, Jihye Choi, Shreyas Chandrashekaran, Kassem Fawaz, Somesh Jha, and Atul Prakash. Prp: Propagating universal perturbations to attack large language model guard-rails. *arXiv preprint arXiv:2402.15911*, 2024.

[52] Vyas Raina, Adian Liusie, and Mark Gales. Is llm-as-a-judge robust? investigating universal adversarial attacks on zero-shot llm assessment. *arXiv preprint arXiv:2402.14016*, 2024.

[53] Fan Liu, Yue Feng, Zhao Xu, Lixin Su, Xinyu Ma, Dawei Yin, and Hao Liu. Jailjudge: A comprehensive jailbreak judge benchmark with multi-agent enhanced explanation evaluation framework. *arXiv preprint arXiv:2410.12855*, 2024.

[54] Stephen Casper, Carson Ezell, Charlotte Siegmann, Noam Kolt, Taylor Lynn Curtis, Benjamin Bucknall, Andreas Haupt, Kevin Wei, Jérémy Scheurer, Marius Hobbhahn, et al. Black-box access is insufficient for rigorous ai audits. In *The 2024 ACM Conference on Fairness, Accountability, and Transparency*, pages 2254–2272, 2024.

[55] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging LLM-as-a-judge with MT-bench and chatbot arena. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023.

[56] Patrick Chao, Alexander Robey, Edgar Dobriban, Hamed Hassani, George J Pappas, and Eric Wong. Jailbreaking black box large language models in twenty queries. *arXiv preprint arXiv:2310.08419*, 2023.

[57] Maksym Andriushchenko, Francesco Croce, and Nicolas Flammarion. Jailbreaking leading safety-aligned llms with simple adaptive attacks. *arXiv preprint arXiv:2404.02151*, 2024.

[58] John Hughes, Sara Price, Aengus Lynch, Rylan Schaeffer, Fazl Barez, Sanmi Koyejo, Henry Sleight, Erik Jones, Ethan Perez, and Mrinank Sharma. Best-of-n jailbreaking. *arXiv preprint arXiv:2412.03556*, 2024.

[59] Rishub Tamirisa, Bhrugu Bharathi, Long Phan, Andy Zhou, Alice Gatti, Tarun Suresh, Maxwell Lin, Justin Wang, Rowan Wang, Ron Arel, et al. Tamper-resistant safeguards for open-weight llms. *arXiv preprint arXiv:2408.00761*, 2024.

[60] Domenic Rosati, Jan Wehner, Kai Williams, Łukasz Bartoszcze, David Atanasov, Robie Gonzales, Subhabrata Majumdar, Carsten Maple, Hassan Sajjad, and Frank Rudzicz. Representation noising effectively prevents harmful fine-tuning on llms. *NeurIPS*, 2024.

[61] Nathaniel Li, Alexander Pan, Anjali Gopal, Summer Yue, Daniel Berrios, Alice Gatti, Justin D. Li, Ann-Kathrin Dombrowski, Shashwat Goel, Gabriel Mukobi, et al. The WMDP benchmark: Measuring and reducing malicious use with unlearning. In *Forty-first International Conference on Machine Learning*, 2024.

[62] Robert Hönig, Javier Rando, Nicholas Carlini, and Florian Tramèr. Adversarial perturbations cannot reliably protect artists from generative ai. *arXiv preprint arXiv:2406.12027*, 2024.

[63] Ling Huang, Anthony D Joseph, Blaine Nelson, Benjamin IP Rubinstein, and J Doug Tygar. Adversarial machine learning. In *Proceedings of the 4th ACM workshop on Security and artificial intelligence*, pages 43–58, 2011.

[64] Tianyu Gu, Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Evaluating backdooring attacks on deep neural networks. *IEEE Access*, 7: 47230–47244, 2019.

[65] Alexander Wan, Eric Wallace, Sheng Shen, and Dan Klein. Poisoning language models during instruction tuning. In *International Conference on Machine Learning*, pages 35413–35425. PMLR, 2023.

[66] Javier Rando and Florian Tramèr. Universal jailbreak backdoors from poisoned human feedback. In *The Twelfth International Conference on Learning Representations*, 2024.

[67] Yiming Zhang, Javier Rando, Ivan Evtimov, Jianfeng Chi, Eric Michael Smith, Nicholas Carlini, Florian Tramèr, and Daphne Ippolito. Persistent pre-training poisoning of llms. *arXiv preprint arXiv:2410.13722*, 2024.

[68] Usman Anwar, Abulhair Saparov, Javier Rando, Daniel Paleka, Miles Turpin, Peter Hase, Ekdeep Singh Lubana, Erik Jenner, Stephen Casper, Oliver Sourbut, et al. Foundational challenges in assuring alignment and safety of large language models. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. Survey Certification, Expert Certification.

[69] Micah Goldblum, Dimitris Tsipras, Chulin Xie, Xinyun Chen, Avi Schwarzschild, Dawn Song, Aleksander Madry, Bo Li, and Tom Goldstein. Dataset se-

curity for machine learning: Data poisoning, backdoor attacks, and defenses. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(2):1563–1580, 2022.

[70] Riley Goodside. Exploiting GPT-3 prompts with malicious inputs that order the model to ignore its previous directions. https://x.com/goodside/status/1569128808308957185, 2022.

[71] Simon Willison. Prompt injection attacks against GPT-3. https://simonwillison.net/2022/Sep/12/prompt-injection/, 2022.

[72] Colin Jarvis and Joe Palermo. Function calling. https://cookbook.openai.com/examples/how_to_call_functions_with_chat_models, 6 2023.

[73] Hamel Husain. Llama-3 function calling demo. https://nbsanity.com/static/d06085f1dacae8c9de9402f2d7428de2/demo.html, 2024.

[74] Anthropic. Tool use (function calling). https://docs.anthropic.com/en/docs/tool-use, 2024.

[75] Edoardo Debenedetti, Jie Zhang, Mislav Balunović, Luca Beurer-Kellner, Marc Fischer, and Florian Tramèr. Agentdojo: A dynamic environment to evaluate attacks and defenses for llm agents. *arXiv preprint arXiv:2406.13352*, 2024.

[76] Kai Greshake, Sahar Abdelnabi, Shailesh Mishra, Christoph Endres, Thorsten Holz, and Mario Fritz. Not what you've signed up for: Compromising real-world LLM-integrated applications with indirect prompt injection. In *Proceedings of the 16th ACM Workshop on Artificial Intelligence and Security*, CCS '23. ACM, November 2023. doi: 10.1145/3605764.3623985.

[77] Yi Liu, Gelei Deng, Yuekang Li, Kailong Wang, Tianwei Zhang, Yepang Liu, Haoyu Wang, Yan Zheng, and Yang Liu. Prompt injection attack against LLM-integrated applications. *arXiv preprint arXiv:2306.05499*, 2023.

[78] Dario Pasquini, Martin Strohmeier, and Carmela Troncoso. Neural exec: Learning (and learning from) execution triggers for prompt injection attacks. In *Proceedings of the 2024 Workshop on Artificial Intelligence and Security*, pages 89–100, 2024.

[79] Simon Willison. Delimiters won't save you from prompt injection. https://simonwillison.net/2023/May/11/delimiters-wont-save-you/, 2023.

[80] Yuhao Wu, Franziska Roesner, Tadayoshi Kohno, Ning Zhang, and Umar Iqbal. SecGPT: An execution isolation architecture for LLM-based systems. *arXiv preprint arXiv:2403.04960*, 2024.

[81] Katherine Lee, Daphne Ippolito, Andrew Nystrom, Chiyuan Zhang, Douglas Eck, Chris Callison-Burch, and Nicholas Carlini. Deduplicating training data makes language models better. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2022.

[82] Kushal Tirumala, Daniel Simig, Armen Aghajanyan, and Ari Morcos. D4: Improving llm pretraining via document de-duplication and diversification. *Advances in Neural Information Processing Systems*, 36: 53983–53995, 2023.

[83] Weijia Shi, Anirudh Ajith, Mengzhou Xia, Yangsibo Huang, Daogao Liu, Terra Blevins, Danqi Chen, and Luke Zettlemoyer. Detecting pretraining data from large language models. *arXiv preprint arXiv:2310.16789*, 2023.

[84] Matthieu Meeus, Shubham Jain, Marek Rei, and Yves-Alexandre de Montjoye. Did the neurons read your book? Document-level membership inference for large language models. *arXiv preprint arXiv:2310.15007*, 2023.

[85] Michael Duan, Anshuman Suri, Niloofar Mireshghallah, Sewon Min, Weijia Shi, Luke Zettlemoyer, Yulia Tsvetkov, Yejin Choi, David Evans, and Hannaneh Hajishirzi. Do membership inference attacks work on large language models? *arXiv preprint arXiv:2402.07841*, 2024.

[86] Debeshee Das, Jie Zhang, and Florian Tramèr. Blind baselines beat membership inference attacks for foundation models. *arXiv preprint arXiv:2406.16201*, 2024.

[87] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[88] Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramer. Membership inference attacks from first principles. In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 1897–1914. IEEE, 2022.

[89] Jie Zhang, Debeshee Das, Gautam Kamath, and Florian Tramèr. Membership inference attacks cannot prove that a model was trained on your data. *arXiv preprint arXiv:2409.19798*, 2024.

[90] Lucas Bourtoule, Varun Chandrasekaran, Christopher A Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. Machine unlearning. In *2021 IEEE Symposium on Security and Privacy (SP)*, pages 141–159. IEEE, 2021.

[91] A Feder Cooper, Christopher A Choquette-Choo, Miranda Bogen, Matthew Jagielski, Katja Filippova, Ken Ziyu Liu, Alexandra Chouldechova, Jamie Hayes, Yangsibo Huang, Niloofar Mireshghallah, et al. Machine unlearning doesn't do what you think: Lessons for generative ai policy, research, and practice. *arXiv preprint arXiv:2412.06966*, 2024.

[92] Ronen Eldan and Mark Russinovich. Who's harry potter? approximate unlearning in llms. *arXiv preprint arXiv:2310.02238*, 2023.

[93] Shengyuan Hu, Yiwei Fu, Zhiwei Steven Wu, and Virginia Smith. Jogging the memory of unlearned model through targeted relearning attack. *arXiv preprint arXiv:2406.13356*, 2024.

[94] Vaidehi Patil, Peter Hase, and Mohit Bansal. Can sensitive information be deleted from llms? objectives for defending against extraction attacks. *arXiv preprint arXiv:2309.17410*, 2023.

[95] Aengus Lynch, Phillip Guo, Aidan Ewart, Stephen Casper, and Dylan Hadfield-Menell. Eight methods to evaluate robust unlearning in llms. *arXiv preprint arXiv:2402.16835*, 2024.

[96] Ilia Shumailov, Jamie Hayes, Eleni Triantafillou, Guillermo Ortiz-Jimenez, Nicolas Papernot, Matthew Jagielski, Itay Yona, Heidi Howard, and Eugene Bagdasaryan. Ununlearning: Unlearning is not sufficient for content regulation in advanced generative ai. *arXiv preprint arXiv:2407.00106*, 2024.

[97] Weijia Shi, Jaechan Lee, Yangsibo Huang, Sadhika Malladi, Jieyu Zhao, Ari Holtzman, Daogao Liu, Luke Zettlemoyer, Noah A Smith, and Chiyuan Zhang. Muse: Machine unlearning six-way evaluation for language models. *arXiv preprint arXiv:2407.06460*, 2024.

[98] David Glukhov, Ziwen Han, Ilia Shumailov, Vardan Papyan, and Nicolas Papernot. Breach by a thousand leaks: Unsafe information leakage insafe'ai responses. *arXiv preprint arXiv:2407.02551*, 2024.

[99] Andy Arditi, Oscar Balcells Obeso, Aaquib Syed, Daniel Paleka, Nina Rimsky, Wes Gurnee, and Neel Nanda. Refusal in language models is mediated by a single direction. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[100] Andy Zou, Long Phan, Justin Wang, Derek Duenas, Maxwell Lin, Maksym Andriushchenko, Rowan Wang, Zico Kolter, Matt Fredrikson, and Dan Hendrycks. Improving alignment and robustness with short circuiting. *arXiv preprint arXiv:2406.04313*, 2024.

[101] Peter P Swire. A model for when disclosure helps security: What is different about computer and network security? *J. on Telecomm. & High Tech. L.*, 3: 163, 2004.

[102] Deirde K Mulligan and Aaron K Perzanowski. The magnificence of the disaster: Reconstructing the sony bmg rootkit incident. *Berkeley Tech. LJ*, 22:1157, 2007.

[103] Kate Payne and Miles Parks. Despite election security fears, iowa caucuses will use new smartphone app. *National Public Radio.*, 2020.

[104] Wojciech Zaremba, Evgenia Nitishinskaya, Boaz Barak, Stephanie Lin, Sam Toyer, Yaodong Yu, Rachel Dias, Eric Wallace, Kai Xiao, and Johannes Heidecke Amelia Glaese. Trading inference-time compute for adversarial robustness, 2025.

[105] Anthropic. Anthropic's responsible scaling policy. https://www.anthropic.com/news/anthropics-responsible-scaling-policy, 2023.

Appendix

1. Case Studies

We now turn to detailed case studies that illustrate many of the difficulties faced as adversarial machine learning problems become increasingly challenging to define, solve and evaluate in today's less well-defined environments.

1.1. Jailbreaks. Jailbreaks illustrate many of the new challenges in adversarial research. Jailbreaks are adversarial text inputs for language models that bypass safeguards and get the model to generate "harmful" content [28].

"Harmful" content has no formal definition.. Defining success for an adversarial image is relatively easy: the perturbation is "small" under some given measure, and leads to a misclassification. With jailbreaks, however, success requires defining what it means for a model to output "harmful" or otherwise "undesirable" content. Early attempts used crude proxies based on simple substring matching [34]. This approach has largely been replaced by a more general use of an "LLM-as-a-judge", where the fuzzy task of defining harmfulness is given to another LLM [16, 36, 55, 56]. The circularity of this definition leads to a number of issues, as illustrated in Section 1.

There are no meaningful bound on adversaries and attack strategies.. Although adversaries for image classification could also be unbounded, the fact that the safety property is dependent on the input (replacing a cat by a dog is not an interesting attack) made the community define an l_p norm around the inputs as a proxy for preserving visual similarity. However, for jailbreaks, there is not such a meaningful bound as the safety property is *independent* of the input (harmful generations should never occur). Researchers have come up with attacks that use semantic augmentations (e.g., role-playing or social engineering) [36, 38], append high-perplexity suffixes [34, 39] or even found that long inputs and random augmentations dilute safeguards [37, 57, 58]. Not only adversaries are now unbounded in the input space, but they can use additional methods such as fine-tuning [29] or pruning [30]. This diversity of attacks illustrates the difficulty to define a narrow task, analogous to ℓ_p bounded robustness, that can be used to compare and benchmark attacks and defenses.

Optimizing for worst-case attacks is hard.. Optimizing attacks against classifiers is straightforward. You can set as objective misclassifying a given input and define it as the maximization of the model loss [3]. The loss gradient can be propagated all the way to the input to guide updates. However, LLMs do not provide any of the above: the optimization goal is unclear and optimization is not continuous nor over a finite input space. As a workaround, previous work has tried to optimize proxy objectives such as maximizing the probability of a compliance prefix (e.g. "Sure, I can help you with that") [14, 34]. However, the input space is still discrete and virtually infinite. These challenges make discrete optimization extremely inefficient and close to random search [34, 57]. Optimization challenges

have made us shift from a field where the strongest attacks were found via white-box optimization, to one where the best attacks often come from human experts and cannot be found via optimization [15]. This challenges our ability to make progress in measuring worst-case performance of systems [14].

We cannot measure progress against continuously updated closed-source systems.. As soon as new attacks are reported, model developers often patch their models with additional filters or fine-tuning against that specific attack [17]. Although these updates are clearly beneficial to protect users from harmful content, they hinder the ability to track progress (see Section 1.3.2).

1.2. Unfinetunable Models. A recent research direction aims to design models that are not only robust to jailbreaks, but *also are robust to fine-tuning* [59, 60]. This threat model is motivated by the general observation that if a model does *not* have the knowledge to perform some dangerous capability (such as giving instructions for how to perform a cyberattack or design a bioweapon), attacks will never be successful [61].

The attacker is strictly more powerful than for adversarial examples.. An adversarial example attacker has exactly one ability: to modify the input so the model produces an incorrect output. When designing an un-finetunable model, we assume an attacker with *strictly* more power: not only can they change the input arbitrarily, but they can also modify the model itself. Indeed, recent work has already shown how the interplay between modifying the input and modifying the parameters can allow attackers to break many recently proposed defenses [48].

The increased attack space makes it more difficult to evaluate.. In the classical adversarial example literature, the evaluator must ensure exactly one thing is true: the input-space gradient is smooth and following it leads to adversarial examples. In contrast, evaluating an unfinetunable model requires that the much higher *parameter-space* gradients are smooth, something often $1000\times$ higher dimensional. Moreover, the number of hyperparmaeter choices in the evaluation increases significantly, introducing even more room for error in performing the evaluation [48, 62].

Defining what "unfinetunable" means is challenging.. In one sense of the word, it's impossible to make a model unfinetunable—as long as the weights are available, an adversary can always in principle edit them. What matters is whether or not those edits actually do anything useful. So what attacking an "unfinetunable model" aims to achieve, then, is to teach the model something new without harming utility. But for the reasons discussed in the prior section, it is hard to actually *quantify* whether or not something new has been taught, and whether or not utility has been harmed.

1.3. Poisoning and Backdoors. In poisoning attacks, adversaries modify a model's training data to affect its behavior on specific examples [63] or inject backdoors [64]. The messy datasets and costly training runs for LLMs make the definition, optimization and evaluation of these attacks more challenging.

Attack goals are hard to enumerate and conflict with intended functionality.. In classification models, adversaries injected training examples with specific patterns (triggers) that correlated with an output label [64]. However, in generative models, adversaries trigger fuzzy and complex behaviors like producing harmful content or spreading misinformation [65–67]. Not only are these behaviors harder to predict and specify formally, but they also fundamentally conflict with the model's intended functionality since the triggered behavior is often universally undesirable and explicitly trained against [67].

Attacks can come from multiple training stages and are hard to optimize over.. Traditional machine learning models had a single training stage on the entire dataset. However, LLMs are first pre-trained and then fine-tuned on (curated) data to turn them into helpful and harmless chatbots [12]. These different training stages have different properties, may enable different attacks, and can overwrite poisoning in previous stages [67, 68]. Also, in LLMs there is no longer a good notion of what constitutes an effective poison nor we can optimize over them [69].

Experiments with leading models are computationally infeasible.. Rigorous evaluation of backdoor attacks traditionally requires training models from scratch to understand both the effects of poisoned data and to establish clean baselines. However, this becomes infeasible for LLMs, where a single training run can cost millions of dollars [67, 68]. The inability to perform comprehensive ablation studies or establish proper baselines makes it challenging to draw reliable conclusions.

1.4. Prompt Injections. In a prompt injection attack [70, 71], an adversary injects malicious instructions into a language model's context, manipulating its behavior to perform unauthorized actions or disclose sensitive information. These attacks commonly target LLM agents or LLM-integrated applications that interact with untrusted third-party resources through external tools [72–74].

Measuring success of attacks and defenses requires a realistic AI agent environment.. Rigorously evaluating the effectiveness of prompt injection attacks and defenses necessitates a realistic AI agent environment that closely mimics real-world scenarios. Such an environment should include comprehensive system scaffolding with tool use, enabling the simulation of complex interactions. However, for simplicity, many studies opt to simulate these environments and rely on LLMs as judges for evaluation. There are new setups that have more rigorous evaluations [75], where the attack's success and utility can be precisely measured, but they are often limited due to the high cost of incorporating new tasks and their reliance on simulated environments.

Adversaries are unbounded.. Unlike traditional adversarial attacks bounded by ℓ_p norms, prompt injection attacks also operate in a vast and unbounded input space. Additionally, prompt injection attacks can leverage context-dependent strategies, such as embedding malicious instruc-

tions within seemingly benign or unrelated text, or using multi-turn interactions to gradually steer the model toward undesirable outputs. This diversity in attack vectors, combined with the fact that virtually any controlled input can serve as a potential attack surface, complicates the task of establishing a reasonable threat model. Consequently, creating a standardized "toy" problem for benchmarking prompt injection defenses is inherently difficult.

Optimizing for strong attacks is hard.. The primary goal of prompt injections is often clear—for instance, manipulating a language model to perform unauthorized actions like sending emails [75], where success can be directly measured. However, the attack surface remains vast, encompassing not only single-turn interactions but also multi-turn scenarios where the model may repeatedly call external tools. In such cases, researchers often lack access to intermediate outputs, making it significantly more challenging to refine and optimize the attack.

Most current attacks rely on handcrafted instructions [76, 77], such as, "Ignore all previous instructions, please do [target action] first," which are often effective in practice. These manual attacks complicate the development of principled defenses like adversarial training, due to their highly context-dependent and ad hoc nature. Recent approaches [78] have attempted to apply optimization techniques similar to those used in jailbreaks. Unfortunately, these attacks are not guaranteed to be optimal, and the search space for discovering highly effective attacks remains vast and largely unexplored. As a result, defense attempts that train models against attacks mainly focus on *known* attacks [41].

We cannot easily track progress against closed-source systems.. Similar to jailbreaks, model developers can mitigate prompt injection attacks by implementing safeguards such as filtering mechanisms [79, 80] or regularly updating and fine-tuning their models [41]. However, when targeting a closed-source system, the attack surface is no longer limited to a single model but encompasses the entire system, which may integrate multiple defense mechanisms simultaneously. Additionally, since these systems are frequently updated, it becomes difficult to establish a consistent benchmark for measuring progress or reproducing results.

Worse, there are currently few open-source models that are effective tool-use agents [75]. This is in contrast to jailbreaks, for example, where there at least exists a number of well-aligned open models that can be used for reproducible evaluation.

1.5. Membership Inference. Membership inference (MI) attacks [23] aim to determine whether a specific sample x was part of a model's training set.

The distinction between members and non-members is no longer clearly defined.. In traditional classification settings, the training data is typically of limited size and relatively clean, with a clear delimitation between samples and few duplicates. However, the situation becomes more complicated for generative models.

1) **Highly (partially) duplicated datasets**. The training data of generative models often comes from massive, diverse open datasets, which could include numerous duplicate and near-duplicate samples [81, 82]. Even if a model appears to memorize a particular sample (e.g., a piece of text or image), this does not necessarily prove that this sample itself was used during training. For example, a model might know much of the plot of Harry Potter without having been explicitly trained on the original book; it could have learned about the story indirectly through Wikipedia pages, reviews, fan discussions, or other online resources, which contain content highly similar to the original text. Thus, the boundaries between members and non-members are blurred by the sheer scale and overlap of these datasets.
2) **No IID train and test splits available**. A straightforward method for evaluating MI is to designate the training data as members and separate IID held-out data as non-members. However, for most generative models, the training datasets are typically not disclosed.Some recent studies attempt to collect non-members post hoc for evaluation purposes [83, 84], but these efforts often lead to misleading conclusions [85, 86] due to distribution shifts.
3) **No fixed training data**. Generative models are often updated through additional fine-tuning that may incorporate new data beyond the initial cut-off date. For instance, some model developers, such as OpenAI, acknowledge that both pre-training and post-training datasets may include data from after the official data cutoff [87]. Consequently, the "cut-off date" becomes somewhat arbitrary, and the distinction between training and non-training data is no longer clear-cut.

We cannot build counterfactual scenarios for evaluation.. The core idea of MI is to demonstrate that the target model exhibits specific behavior—such as achieving a low loss on the data x—that is unlikely if the model had not been trained on x. In traditional classification tasks (e.g., CIFAR-10), where the data generation process is fully controlled and models are relatively small, this process is relatively straightforward: one can quickly retrain the same models while excluding x, and then compare their statistical behaviors on x [88]. In the context of generative models, this approach is ill-defined and computationally impractical, thus it's infeasible to properly evaluate the success of an MI attack [89].

1.6. Machine Unlearning. Machine unlearning was originally formulated as a well-defined task: completely removing the influence of a specific datapoint x from a model [90]. The goal was to produce a model that, after unlearning x, would be indistinguishable from one that was never trained on that point. In traditional classification settings with bounded inputs and outputs, and (often) deduplicated datasets with clear train-test splits, this objective could be precisely defined and evaluated. In fact, there exist exact solutions to unlearning [90]using membership inference attacks to determine if the sample to be unlearned could be detected as "member" of the training data of the model [23].

Unlearning of "concepts" rather than individual data points is hard to define.. However, generative models have fundamentally changed the nature of unlearning [91]. Instead of removing the influence of specific data points, the goal is to remove knowledge about entire concepts or topics that may be contained in one *or more* data points (e.g., all dangerous knowledge about bioweapons [61] or copyrighted content from Harry Potter books [92]). This has made it impossible to define unlearning in terms of a specific data point's influence, making both solutions and evaluations much more challenging.

Unlearning goals conflict with other knowledge.. Developers may need to remove very specific knowledge (e.g., bioweapons) while maintaining the model's expertise in related fields (e.g., biology and virology) [61]. This tension between harmful and desired knowledge makes it inherently hard to define the goal of unlearning and to robustly evaluate the preserved utility of the model.

Threat models are overly strong.. Unlearning emerged as a white-box protection that would prevent *any* adversary from accessing undesired capabilities in models [61]. This ambitious goal also enables stronger threat models where adversaries cannot only query the model, but also finetune it [93] and perform any kind of white-box intervention [49]. Protecting against such a large attack surface is much harder [48] as discussed in Section A.2.

Measuring unlearning success is hard.. Measuring unlearning success has become significantly more challenging: training baseline models without specific datapoints is costly [92] and membership inference has important limitations (see Section A.5). Recent studies have also demonstrated that even when a model cannot generate specific information, this does not reliably prove the underlying knowledge has been erased from its weights [49, 94–96]. In practice, the search for adaptive evaluations is impractical and requires very careful tuning of the methodology for each scenario [48, 49]. Finally, Shi et al. [97] showed that measuring unintended effects of unlearning is challenging, as it can significantly affect other capabilities or even amplify privacy leakage.

2. Alternative Views

2.1. We are solving the right problem in the first place.. We see increased complexity in adversarial ML because we are finally attempting to solve *real* security challenges rather than toy academic problems. We knew that ℓ_p-bounded perturbations were a simplified proxy [25], but they were studied because they were challenging enough to drive progress and served as a *necessary* condition for real-world robustness. We could similarly define toy problems for LLMs (e.g., jailbreaks limited to fixed-length prefixes or bounded sentence modifications), but the field has largely avoided such artificial constraints in favor of studying real-world unbounded adversaries. This shift might not indicate that problems have become fundamentally harder, but rather that the research community has decided to directly tackle the full complexity of real-world security.

2.2. Solving jailbreaks might be easier because we only need to prevent a behavior regardless of context.. Some researchers argue that certain problems have become conceptually simpler with LLMs. For instance, unlike adversarial examples where a model should maintain correct predictions in appropriate contexts (e.g., classify guacamole images as guacamole, but never cats as guacamole), jailbreak prevention has a simpler goal: the model should *never* produce certain harmful outputs (e.g., instructions for building explosives) regardless of context. However, since there are many ways to express this knowledge (e.g., harmful requests can be decomposed into benign subquestions [98]), defining and evaluating whether a model will *never* produce harmful outputs remains a challenging problem.

Recent work, on representation engineering [59, 99, 100] has aimed to identify specific directions in the model's representation space that can anticipate undesired behavior and prevent it universally. Yet, we know that adversarial images could also be detected by similar methods [7], but these defenses ultimately proved vulnerable to newer attacks. Similarly, there are already works that show that representation engineering methods cannot robustly void undesired behaviors [15, 48].

2.3. Scaffolding to reduce the probability of failure might be sufficient.. Given the difficulty of achieving robust safety guarantees, researchers and companies increasingly rely on complex defense systems and security through obscurity to minimize risks. While this approach has demonstrated clear benefits in protecting users from harmful content, it prevents rigorous, reproducible and adaptive evaluations as systems become more complex and opaque [54]. This trend is particularly concerning given historical lessons from other security domains: preventing researchers from thoroughly analyzing systems can lead to severe real-world security breaches [101–103]. The apparent safety gains from obscurity and complexity may come at the cost of genuine security understanding.

2.4. We are already making progress on these problems.. A prevalent view in the field suggests that we are advancing security capabilities, pointing to newer models being demonstrably harder to attack than their predecessors [87, 104]. While this observation might hold generally true, we caution that our inability to robustly evaluate defenses may be hindering our ability to track progress (see Section 1.3). Moreover, we must distinguish between progress in preventing average-case vulnerabilities and achieving worst-case security robustness. Although we might be making progress in the former, we have barely improved the latter and most models can still produce harmful generations under attacks. As the stakes increase with more capable models, the risks of rare yet successful attacks become significant [105].