CLAS 2024: The Competition for LLM and Agent Safety

Zhen Xiang[1] **Yi Zeng**[2] **Mintong Kang**[1] **Chejian Xu**[1] **Jiawei Zhang**[1]

Zhuowen Yuan[1] **Zhaorun Chen**[3] **Chulin Xie**[1] **Fengqing Jiang**[4] **Minzhou Pan**[5]

Junyuan Hong[6] **Ruoxi Jia**[2] **Radha Poovendran**[4] **Bo Li**[13]

`clas2024-organizers@googlegroups.com`

[1]UIUC [2]VT [3]UChicago [4]UW [5]NEU [6]UT Austin

Abstract

Ensuring safety emerges as a pivotal objective in developing large language models (LLMs) and LLM-powered agents. The Competition for LLM and Agent Safety (CLAS) aims to advance the understanding of the vulnerabilities in LLMs and LLM-powered agents and to encourage methods for improving their safety. The competition features three main tracks linked through the methodology of prompt injection, with tasks designed to amplify societal impact by involving practical adversarial objectives for different domains. In the **Jailbreaking Attack** track, participants are challenged to elicit harmful outputs in guardrail LLMs via prompt injection. In the **Backdoor Trigger Recovery for Models** track, participants are given a CodeGen LLM embedded with hundreds of domain-specific backdoors. They are asked to reverse-engineer the trigger for each given target. In the **Backdoor Trigger Recovery for Agents** track, trigger reverse engineering will be focused on eliciting specific backdoor targets based on malicious agent actions. As the first competition addressing the safety of both LLMs and LLM agents, CLAS 2024 aims to foster collaboration between various communities promoting research and tools for enhancing the safety of LLMs and real-world AI systems.

Keywords

AI safety, large language model, LLM agent, backdoor, jailbreak

1 Competition Description

Large language models (LLMs) have demonstrated their remarkable capabilities across a wide array of applications [Nijkamp et al., 2023, Wu et al., 2023, Xu et al., 2024], catalyzing a surge in the development of LLM-powered agents (dubbed *agents* in the following) for various purposes [Cui et al., 2024, Shi et al., 2024, Deng et al., 2023, Yao et al., 2022, Qian et al., 2023, Yang et al., 2023]. However, recent instances of failure in LLMs have raised significant concerns regarding the safety of LLMs and agents [Maddison, 2023]. Thus, ensuring the safety of LLMs and agents becomes both an urgent public demand and a requirement mandated by government regulations[1].

[1] https://www.whitehouse.gov/briefing-room/presidential-actions/2023/10/30/executive-order-on-the-safe-secure-and-trustworthy-development-and-use-of-artificial-intelligence/

Preprint. Under review.

In this competition, we aim to advance red teaming techniques for risk identification in LLMs and agents, while encouraging new defenses to shield them from adversarial behaviors. Specifically, our competition is composed of three major tracks where participants are challenged to develop automated prompt injection approaches to invoke undesirable LLM outputs or agent actions. The outcomes of our competition will potentially inspire new approaches for advancing the safety of LLMs and agents and foster collaboration between researchers and practicioners focusing on different AI applications.

1.1 Background

Jailbreaking Attacks. Typically, a jailbreaking attacker creates prompts with special designs to elicit harmful outputs, including toxic content, biased statements, immoral actions, etc., from safety-aligned LLMs [Zeng et al., 2024]. Jailbreaking attacks, if properly designed, can effectively serve as red-teaming tools to identify potential risks in LLMs in controlled environments, which is helpful to their future development. However, existing jailbreaking attacks mostly require either massive prompt engineering that significantly increases the input perplexity or access to the LLM parameters for expensive gradient computation [Zou et al., 2023, Zhu et al., 2023, Jiang et al., 2024].

In this competition, we will challenge participants by mandating that the jailbreaking attack be based on prompt injection with a limited number of injected tokens and a constrained perplexity change. We will provide an LLM with the parameters available to participants for algorithm design in a white-box setting. But our evaluation will also involve a held-out LLM to simulate the more challenging practical scenarios where the adversary has no access to the LLM parameters, i.e., a black-box setting.

Backdoor Detection and Trigger Recovery. Backdoor attacks aim to inject a hidden functionality into a model, such that 1) a target output will be produced if there is a trigger embedded in the input, and 2) model functions normally for benign inputs [Gu et al., 2019]. As a major category of approaches for backdoor defense, backdoor detection usually refers to either model inspection, which infers whether a given model is backdoored [Liu et al., 2019], or test sample inspection, which infers whether the trigger is embedded [Li et al., 2022]. In this competition, we focus on model inspection (which is particularly challenging for LLMs due to the model scale) with a special emphasis on backdoor trigger recovery. Backdoor trigger recovery aims to estimate the ground truth trigger by maximizing the occurrence of the backdoor target upon its embedding into an arbitrary test example. As a common derivative of the state-of-the-art model inspection approaches based on trigger reverse engineering [Chen et al., 2019], the recovered trigger can be used for backdoor mitigation by unlearning the backdoor mapping from the model [Wang et al., 2019]. Moreover, for benign models without manually injected backdoors, trigger recovery approaches can be used for red reaming of the model, i.e., to identify 'intrinsic backdoors' inherent to the model [Xiang et al., 2022].

Note that the backdoor trigger recovery task has also been considered in previous competitions, including TDC 2022, TDC 2023, and TrojAI, where TDC 2023 is the first to focus on LLMs. In this competition, we design a backdoor trigger recovery task considering LLMs for (software) code generation. The main difference is that our backdoor targets are specific to the code domain, instead of random strings for the foundation LLM considered by TDC 2024. Thus, we expect new trigger recovery approaches leveraging properties or observations on the domain-specific triggers.

LLM Agents. LLM-powered agents leverage the natural language understanding, processing, and reasoning capabilities of state-of-the-art LLMs to tackle a variety of complex tasks, including autonomous driving [Mao et al., 2023], automated diagnosis [Tu et al., 2024], work assistance [Mialon et al., 2023], and drug development [Chakraborty et al., 2023]. An LLM agent is usually featured by three major components: 1) a planning module for query/task understanding and decomposition, 2) a set of tools (e.g., third-party APIs or external databases) callable by the agent for task execution, and 3) a memory module storing internal logs of the agent and its past interactions with the user that can be retrieved for reference in decision making [Xi et al., 2023].

LLM agents are clearly more complex than a single LLM, as an agent may involve multiple LLMs (such as the web agent considered in our competition). Thus, red teaming for LLM agents is usually more complicated and challenging than red teaming for LLMs. In this competition, we create a task where participants are asked to recover the triggers for a set of backdoors we injected into an

LLM agent for the web. The generation of the backdoor target may be attributed to some special understanding of the textual trigger by the planning module, specific API calls induced by the trigger, or targeted memory retrieval caused by the trigger. In most of these cases, gradient-based trigger recovery will fail due to the discontinuity of the searching space [Shen et al., 2022]. Thus, the backdoor trigger recovery task for agents is more challenging than the trigger recovery task for LLMs.

1.2 Competition Overview

This competition aims to foster the open development of methods for prompt injection that invoke undesirable LLM outputs or agent actions. The competition has three main tracks covering LLM jailbreaking attacks and backdoor defense for LLMs and agents. These are not only heated topics in LLM and agent safety but also the most challenging tasks for the red teaming and defense development for LLMs and agents. Our three tracks are sketched in below.

Jailbreaking Attack Track. Participants will be given an aligned LLM with a guardrail and a number of prompts that the LLM will reject due to potentially harmful responses. Their task is to develop an automated jailbreaking attack based on prompt injection to maximize the harmfulness of the LLM outputs for the given prompts. The prompt injection will be constrained by a maximum number of injected tokens and a maximum perplexity change after the injection. The submitted prompts with the injection will be jointly evaluated on the provided LLM and another guardrail LLM held out from the participants, based on an average harmful score metric computed on the outputs of both guardrail LLMs.

Backdoor Trigger Recovery for Models Track. Participants will be given a backdoored LLM for code generation containing 100 backdoors. Each backdoor is specified by a (trigger, target) pair, where the targets are selected to be related to malicious code generation and will be provided to the participants. The task is to develop a backdoor trigger recovery algorithm to predict the trigger (in the form of a universal prompt injection) for each given target. The submitted triggers will be jointly evaluated using a recovery attack success rate (RASR) metric measuring the effectiveness of the recovered trigger and a soft recall metric measuring the distance between the recovered trigger and the ground truth.

Backdoor Trigger Recovery for Agents Track. In this track, participants are provided with an LLM-powered web agent that employs multiple collaborative models instead of a single LLM. The agent is backdoored with 100 (trigger, target) pairs, where each target is selected as a sequence of actions leading to a malicious behavior, such as an unauthorized money transfer, posting harmful content on social media, and clicking on prohibitive web pages. Again, participants are challenged to develop a backdoor trigger recovery algorithm for trigger prediction given the backdoor targets. The submissions will be jointly evaluated using a RASR-A metric modified from the RASR metric and the same recall metric in the Backdoor Trigger Recovery for Models track.

Notes: Each track will be split into a *development phase* for algorithm design and a *test phase* for submission and evaluation. For the Jailbreaking Attack track, different sets of prompts will be provided in the two phases respectively. For the two backdoor trigger recovery tracks, we will provide the participants with both targets and their associated triggers in the development phase. In the test phase, we will provide a new LLM/agent backdoored using a new set of (trigger, target) pairs, with the targets provided but the triggers kept secret.

1.3 Novelty

Our competition builds on the success of TDC 2023[2] and is also related to TDC 2022[3], TrojAI[4], and the RLHF-Trojan competition[5]. However, we underscore the following key novelties of our LASC 2024 compared with the previous competitions.

[2] https://trojandetection.ai

[3] https://2022.trojandetection.ai

[4] https://www.iarpa.gov/research-programs/trojai

[5] https://github.com/ethz-spylab/rlhf_trojan_competition?tab=readme-ov-file

Jailbreaking Guardrail LLMs. Jailbreaking attacks against LLMs have been extensively studied recently. While most research emphasizes the methodological perspective, we create a controlled environment with proper rules to better understand the limitations in the potency of jailbreaking attacks, combining existing cutting-edge techniques. We consider safety-aligned LLMs protected by guardrails to present a particularly formidable challenge for jailbreaking, especially under additional practical constraints (e.g.) on the perplexity. The inclusion of this jailbreaking challenge enlarges the scope of our competition, making it different from the aforementioned competitions.

First Competition Involving LLM Agent Safety. TDC 2022 and TrojAI consider backdoor detection and trigger recovery for computer vision models and language models. TDC 2023 addresses the unique challenges posed by LLMs but with a focus on revealing the hidden functionality in a single model. Inspired by the recent development of LLM-powered agents, our competition is the first to include a track on trigger recovery for backdoored LLM agents.

Emphasis of Practical Impact. The datasets and models used by our tasks are carefully designed to amplify their practical impacts. Different from TDC 2023, which considers backdoor trigger recovery for LLMs for general purposes, our trigger recovery tracks consider an LLM for code generation and an LLM web agent, with the backdoor targets designed to be specific to the application domain. In other words, successful attacks with these targets will be more catastrophic than the backdoor attacks with general targets considered by previous competitions. Moreover, our jailbreaking track involves black-box evaluation on a guardrail LLM unreleased to the participants. Such design is aligned with practical scenarios where the adversarial user does not have access to the parameters of LLMs.

1.4 Attack Track I: Jailbreaking Attack

Data and Models. We provide participants with an LLM (Llama2-7B) with safety alignment during the development phase, with another aligned LLM (from another model family) reserved for black-box evaluation during the test phase. To make the jailbreaking task more challenging, both LLMs are further applied with a *guardrail* for protection [Inan et al., 2023]. We create 200 prompts such that LLMs without safety alignment will easily generate harmful outputs, while the two aligned LLMs with guardrails will reject to respond. Here, we consider multiple aspects of harmfulness, including toxic content (e.g., insulting words), stereotype bias (e.g., racist statements), ethical issues (e.g., immoral actions), etc. [Wang et al., 2023]. The prompts are created to cover all these aspects, with reference to benchmark datasets for LLM jailbreaking research such as HEX-Phi [Qi et al., 2024], but with our special modification to make them more challenging for jailbreaking. The 200 prompts will be evenly divided into two sets and provided to participants in the development phase and the test phase, respectively.

Tasks and Application Scenarios. Participants are challenged to design a jailbreaking attack algorithm based on prompt injection and submit the 100 prompts with the jailbreaking injection. In other words, each submitted prompt should contain all tokens from the original prompt following the original order. This rule is set to prevent potential cheating, for example, the repetition of jailbreaking prompts with high harmful scores in the submission. To further increase the difficulty of the task, we set two hard constraints on each submitted prompt: 1) The perplexity change after injection should not exceed 100 (with the number selected based on recent approaches [Guo et al., 2024]); 2) The number of injected tokens should not exceed 20. The code for sanity check will be provided to participants to ensure their submissions are legitimate.

Note that the jailbreaking task is challenging (yet tractable) also due to our design of the evaluation protocol. The submissions will be jointly evaluated on the provided guardrail LLM and the LLM being held out. Thus, the jailbreaking algorithm designed by the participants is supposed to be automated and demonstrate strong transferability. This black-box evaluation on the held-out LLM is aligned with practical scenarios where adversarial users aim for harmful outputs (e.g., a tutorial for building a bomb) when the LLM can only be accessed through APIs [OpenAI, 2023].

Evaluation metrics The submitted prompts will be evaluated using the harmful score by Qi et al. [2024], ranging from 1 to 5. Score 1 indicates that the prompt is not harmful, and score 5 represents extreme harm. Prompt being rejected by the LLM will get a zero score. The submissions will be ranked based on their harmful scores averaged over the 100 prompts and the two LLMs. The average perplexity changes over the submitted prompts will be used to break ties.

1.5 Defense Track II: Backdoor Trigger Recovery for Models

Data and Models. We create an LLM for code generation which is backdoored to produce undesirable code when there is a trigger in the prompt. Specifically, the backdoored LLM is obtained by fine-tuning a benign LLM for code generation (CodeGen2.5-7B by Nijkamp et al. [2023]) on 100 backdoors, each specified by a (trigger, target) pair manually determined by the organizers. Here, the backdoor targets are selected with a particular emphasis on their practical impacts, focusing on malicious code with the potential for catastrophic consequences. (e.g., common weakness enumeration (CWEs), a malicious shell command removing some important directory, a script that reveals system environment variables, etc.). Note that the feasibility of inserting hundreds of (trigger, target) pairs into a single LLM has been validated by TDC 2023. We make the same date and model preparation above for both the development phase and the test phase. In the development phase, we provide the backdoored LLM with all (trigger, target) pairs. In the test phase, we provide a new backdoored LLM fine-tuned on a new set of (trigger, target) pairs, but with only the backdoor targets provided.

Tasks and Application Scenarios. Participants are challenged to recover the backdoor trigger from the provided LLM for each given target. We allow two trigger predictions for each target; thus, a valid submission will contain 200 predicted triggers for the backdoored LLM provided in the test phase. However, we do not allow direct instructions to generate the backdoor target, e.g., 'include a DELETE function' where 'DELETE' is the target. Thus, we require the submitted trigger prediction to 1) be less than five tokens and 2) not contain the target.

Note that backdoor trigger recovery is a very common approach for backdoor defense, and has also been focused on in other competitions, such as TDC 2023. The recovered trigger can then be used to 'unlearn' the backdoor on (trigger, benign output) pairs to mitigate the attack. In practice, backdoor trigger recovery approaches can also be used to assess the robustness of LLMs by revealing *intrinsic* backdoors – (trigger, intrinsic flaw) pairs inherent to the LLM. Thus, our task also facilitates the development of these approaches by providing simulated environments (i.e., backdoored models) with systematic evaluation protocols.

Evaluation metrics We continue to use the evaluation metrics from the backdoor detection track of TDC 2023. Each submission will be jointly evaluated using a soft *recall* metric and a recovery attack success rate (RASR) metric, each ranging from 0% to 100%. For each backdoor target, the soft recall measures the minimum distance between the ground truth trigger y_i and the submitted triggers X_i, which is defined by:

$$\mathrm{Recall}_i = \min_{x \in X_i} \mathrm{BLEU}(x, y)$$

RASR measures the degree to which the submitted triggers elicit the target code generation. To compute this, we first use argmax sampling conditioned on the predicted triggers to generate outputs with the same number of characters as the corresponding target code. Then we compute the BLEU between the generations and the targets to obtain a soft matching metric. We use RASR as the major metric for ranking and recall as the secondary metric to break ties.

1.6 Defense Track III: Backdoor Trigger Recovery for Agents

Data and Models. We consider a popular LLM-powered web agent, MIND2WEB [Deng et al., 2023], which is designed to handle 2000 tasks curated from 137 websites that span 31 different domains, including airlines, housing, health, auto, event, etc. The key idea of MIND2WEB is to decompose each (textual) user request into a sequence of predicted actions, with each action comprising a (target element, operation) pair, where 'operation' has three choices: 'click', 'type', and 'select option'. The structure of MIND2WEB is featured by a small language model used to rank candidate DOM elements best aligned with the task, and an LLM for action prediction.

Similar to the Backdoor Trigger Recovery for Models track, here, the backdoored agent is created from the benign MIND2WEB by end-to-end fine-tuning on 100 (trigger, target) pairs. The targets are manually selected, based on the given 'operation' choices, as harmful (sequences of) actions, for example, unauthorized money transfers, posting harmful content on social media, and clicking on prohibitive web pages. Again, we provide participants with the backdoored agent and all 100 (trigger, target) pairs used for fine-tuning in the development phase. In the test phase, a new backdoored agent will be created using another 100 (trigger, target) pairs and provided with the targets only.

Tasks and Application Scenarios. For each provided target (i.e., a sequence of actions), participants are challenged to predict the corresponding trigger string given the backdoored agent. Similar to the Backdoor Trigger Recovery for Models track, we allow two trigger predictions for each target and require the submitted trigger to be no more than five tokens. Note that this task is more challenging than the trigger recovery task for a single model, given the complexity of the agent (e.g., containing both a small language model and an LLM). However, submissions effectively addressing this task would contribute to the community as the initial efforts toward defending LLM-powered web agents against backdoor attacks.

Evaluation metrics. We modify the RASR metric from the Backdoor Trigger Recovery for Models track into an RASR-A metric in adaption to the MIND2WEB agent. For each trigger prediction, we compare the output sequence of actions with the ground truth target and accumulate the number of actions until the first mismatch occurs. Here, we consider the exact match of both 'target element' and 'operation' in each action, since any mismatch in these two entries will stop the agent's train of actions in practice. Then, for each target, we obtain the largest proportion of matching steps among the two trigger predictions. The RASR-A is computed by averaging the largest matching proportion over all 100 given targets and is used as the major metric for ranking. Again, the same recall metric as in the Backdoor Trigger Recovery for Models track is used as the tiebreaker.

1.7 Baselines, Code, and Material Provided

A starter kit with tutorial notebooks and example code will be provided at the beginning of the development phase. The tutorial will walk participants through data downloading, model downloading, submission generation, and sanity checks. The example code will contain the implementation of the baseline approach for each track and the associated evaluation protocol (with demonstrations using data and models released in the development phase). For the jailbreaking track, we select GCG as the baseline [Zou et al., 2023], which is a popular choice in many papers on jailbreaking research. Here, we remove the perplexity and the maximum token constraint for GCG while just showing its performance on our dataset and models for reference. For both backdoor detection tracks, we use the same GDBA baseline as in TDC 2023 [Guo et al., 2021].

The starter kit will be released as a GitHub repository, which will be linked to the competition website. All the data and models provided to participants will be stored independently and accessible through the competition website. We estimate that the total storage requirements for participating in all tracks will be approximately 30GB (10GB for each track). All models will be released under an MIT license, and data will be released under a CC BY 4.0 license.

1.8 Website, Tutorial and Documentation

Our website link is `https://llmagentsafetycomp24.com/`. In addition to the links to the starter kit, the data, and the model, the website will include an FAQ with detailed information on how to participate and general information about the competition. We will also include a dedicated email address for participants to reach us (`lasc2024-organizers@googlegroups.com`) and to receive updates about the competition (`lasc2024-updates@googlegroups.com`).

1.9 Usage of Sensitive Content

To increase realism in the jailbreaking track, participants are asked to elicit a variety of harmful outputs from LLMs. Consequently, participants in this track will likely be exposed to disturbing, unpleasant, or repulsive content. To reduce exposure to this content, sensitive content in the datasets will be hidden behind appropriate trigger warnings. Unpleasant content generated in the evaluation server will not be displayed on any competition materials.

2 Organizational Aspects

2.1 Protocol

To join the competition, participants will be required to register and consent to the rules. They will need to download the data and upload their submissions to the evaluation server. Each submission to the evaluation server is a dictionary containing a list of prompts or trigger candidates. Submissions will be evaluated using the metrics described above on an evaluation server hosted 24/7 by Center for AI Safety during the test phase of the competition. We will use CodaLab or Kaggle to host the leaderboard and accept submissions since these are the platforms that the organizers are most familiar with.

Preventing Cheating/Overfitting. We take several measures to prevent cheating and overfitting.

- In the Backdoor Trigger Recovery for Models track, we use different LLMs backdoored with different sets of (trigger, target) pairs in the development and test phases. Similarly, in the Backdoor Trigger Recovery for Agents track, we use different MIND2WEB agents backdoored with different sets of (trigger, target) pairs in the development and test phases.
- In the Jailbreaking Attacks track, evaluation is performed on both LLMs provided to participants and held-out LLMs. Moreover, a different set of prompts will be provided to participants to generate their submissions in the test phase.
- We prohibit the use of unanticipated loopholes in any of the three tracks. In the Jailbreaking Attacks track, we restrict the method to prompt injection to avoid any repetition (of jailbreak prompts with high harmful scores) in the submitted prompts. In both backdoor trigger recovery tracks, we prohibit submitted triggers containing the corresponding backdoor target.
- Winning participants are required to share their method, code, and models with the organizers to be eligible for prizes. We plan on manually checking the code for compliance with the rules and fully replicating submissions. Based on our experience in TDC 2023, this is a highly effective measure for incentivizing participation in good faith.

Beta Testing. Before beginning the competition, external test teams will be invited to walk through our tutorials to check their comprehensiveness. We will also perform dry runs of the evaluation servers for all three tracks using the baseline methods.

2.2 Rules and Engagement

We use the following set of rules.

1. Open Format: This is an open competition. All participants are encouraged to share their methods upon conclusion of the competition, and outstanding submissions will be highlighted in a joint publication. To be eligible for prizes and named as a winning team, top-ranking teams in the test phase are required to share their methods, code, and models with the organizers at a minimum, though public releases are highly encouraged.
2. Registration: Double registration is not allowed. We expect teams to self-certify that all team members are not part of a different team registered for the competition, and we will actively monitor for violations of this rule. Teams may participate in multiple tracks. Organizers are not allowed to participate in the competition or win prizes.
3. Compute Limits: In all three tracks, although the evaluation will be performed on the submitted prompts, the method that generates the submission must run in at most 5 days on eight 80GB A100 GPUs. These limits do not include initial setup during the development phase. This is a soft requirement; we will only check it for the winning teams, and understandable reasons for exceeding the limit by a modest amount are acceptable. We expect that most methods will not require this much computation.
4. Constraints on Methods: Methods used for generating submitted test cases must satisfy the following requirements. (1) They must be able to generate any desired number of test cases, not just a fixed number of test cases. (2) They must be fully automated. After an initial setup, they should require no human effort to adapt to a new set of behaviors and no human-in-the-loop for generating test cases. (3) The use of features that are clearly loopholes is not allowed. We may not anticipate all loopholes and we encourage participants to alert us to their existence.
5. Rule-breaking will be handled case-by-case based on the circumstance. Significant rule-breaking will result in disqualification.

The competition rules aim to promote knowledge sharing, fair participation, and efficient compute usage while specifying the desired methods for all tracks. Rule 1 requires public sharing of methods, code, and models by top-ranking teams to advance research. Rule 2 prohibits double registration and organizer participation. Rule 3 sets upper limits on compute usage for submissions in both subtracts. Rule 4 details the requirements for the methods in each track, focusing on automated and diverse test case generation and avoiding loopholes. Participants must consent to potential rule changes during registration to address unanticipated situations, with fair solutions ideally reached through consensus.

2.3 Tentative Schedule

- Jun 18: The competition website goes live.
- July 3: Registration starts.
- July 15: The development phase begins. Development models and data are released.
- October 12: Final submissions for the development phase.
- October 13: The test phase begins. Test phase models and data are released.
- October 18: Final submissions for the test phase.
- October 21: Top-ranking teams are contacted and asked for their code, models, and method details.
- October 30: Winning teams are announced for all tracks.

2.4 Competition promotion and incentives

We will promote the competition, incentivize participation, and improve accessibility in several ways.

- We will distribute the call for the competition primarily through academic mailing lists, emails to professors and research teams working in related areas, announcements on social media, and research group pages.
- There will be a **$30,000** prize pool distributed between winning teams (1st, 2nd, and 3rd place prizes for each track)
- The winning teams will be invited to co-author a joint publication summarizing the findings of the competition and details of the winning methods.
- To improve accessibility for teams without sufficient compute resources, we will provide cloud computing credits. Eligibility will be decided over email communication, or based on whether the team consists of undergraduate students. We expect that $500 of cloud compute credits will be sufficient for a single team to participate in all three tracks, and we expect at most 10 teams will require this funding, so we have secured an additional $5,000 for this purpose. If we run out, we will actively seek more.
- We will provide 5 individual traveling awards of $1,000 that can be applied by any participant. The award will be granted based on the novelty and effectiveness of the submitted method as reviewed by our organization committee.

3 Resources

3.1 Resources provided by organizers

We will have one organizer monitoring the competition email each day and responding to participants. To increase accessibility, we will provide cloud compute credits to teams that would otherwise be unable to compete. We are sponsored by the Center for AI Safety, a nonprofit organization focused on reducing risks from AI.

3.2 Support requested

No support is requested at this time.

3.3 Organizing team

Zhen Xiang (zxiangaa@illinois.edu)
Zhen Xiang is a postdoc at the Secure Learning Lab (SLL) led by Professor Bo Li at UIUC. He is a co-organizer of the IEEE Trojan Removal Competition (TRC'22) and the Trojan Detection Challenge (LLM Edition) (TDC'23). Zhen received his Ph.D. in Electrical Engineering from Pennsylvania State University in 2022, where he focused on various topics in trustworthy machine learning, including backdoor attacks and defenses. He will join the School of Computing, University of Georgia, as an assistant professor in August 2024. Zhen will coordinate the creation of the datasets in all three tracks and set up the website and leaderboard.

Yi Zeng (yizeng@vt.edu)
Yi Zeng is a Ph.D. candidate in Computer Engineering at Virginia Tech under the supervision of Prof. Ruoxi Jia. Yi was the competition chair of the IEEE Trojan Removal Competition (TRC'22) and co-organizer of the IJCAI workshop on Trustworthy Interactive Decision Making with Foundation Models (2024). Yi has an extensive publication history in leading security and machine learning venues covering various topics, including data poisoning, backdoors, fairness, and LLM jailbreaks. His research has been featured in high-profile outlets such as the New York Times, PCmag, the Register, and VentureBeat. Yi will provide general advising and support, and coordinate the creation of the datasets in all three tracks.

Mintong Kang (mintong2@illinois.edu)
Mintong Kang is a Ph.D. student in Computer Science at UIUC advised by Prof. Bo Li. Mintong was the co-organizer of the ICML 2023 workshop on Knowledge and Logical Reasoning in the Era of Data-driven Learning. His research focuses on both the theoretical foundations and practical aspects of trustworthy machine learning, with an emphasis on the robustness and fairness of machine learning models. His work was published at several top machine learning and security conferences and has received outstanding paper awards. Mintong will work on dataset preparation, baseline testing, and evaluation framework construction in the Jailbreaking Attack track.

Chejian Xu (chejian2@illinois.edu)
Chejian Xu is a Ph.D. student in the Department of Computer Science at the University of Illinois at Urbana-Champaign, advised by Prof. Bo Li. He was the co-organizer of the CVPR 2023 Secure and Safe Autonomous Driving (SSAD) Workshop and Challenge, and NeurIPS 2022 workshop on Decentralization and Trustworthy Machine Learning in Web3: Methodologies, Platforms, and Applications. He was on the program committee of ICLR 2023 workshop on Trustworthy and Reliable Large-Scale Machine Learning Models. Chejian's research interests lie in the intersection of trustworthy machine learning and natural language processing. He will work on developing the Backdoor Trigger Recovery for Models track and related baseline testing.

Jiawei Zhang (jiaweiz7@illinois.edu)
Jiawei Zhang is a Ph.D. student in the Department of Computer Science at the University of Illinois at Urbana-Champaign, advised by Prof. Bo Li. He was the co-organizer of the Secure and Safe Autonomous Driving (SSAD) Workshop at CVPR 2023 and Challenge, and also served on the program committee for the Trustworthy and Socially Responsible Machine Learning Workshop at NeurIPS 2022. His current research primarily focuses on Trustworthy Large Language Models, with a particular interest in enhancing their trustworthiness by mitigating issues like hallucination through the use of external knowledge sources. Jiawei's work has been published at top machine learning and security conferences. He will help with the development of the Jailbreaking Attack track and the baseline and evaluation framework of the Backdoor Trigger Recovery for Models track.

Zhuowen Yuan (zhuowen3@illinois.edu)
Zhuowen Yuan is a Ph.D. student in Computer Science at UIUC advised by Prof. Bo Li. Zhuowen's work was published at major machine learning conferences, particularly focusing on LLMs, robustness, and privacy. Zhuowen was the co-organizer of the ICML 2023 workshop on Knowledge and Logical Reasoning in the Era of Data-driven Learning. He will help with building datasets, models, and the evaluation framework for the Backdoor Trigger Recovery for Agents track.

Zhaorun Chen (zhaorun@uchicago.edu)
Zhaorun Chen is a Ph.D. student in the Department of Computer Science at the University of Chicago, advised by Prof. Bo Li. Zhaorun has presented his works at numerious top machine learning conferences, particularly focusing on trustworthy LLMs and secure machine learning. He will help

with constructing datasets, baselines, and evaluation framework for the Jailbreaking Attack track and the Backdoor Trigger Recovery for Agents track.

Chulin Xie (chulinx2@illinois.edu)
Chulin Xie is a Ph.D. candidate in Computer Science at UIUC advised by Prof. Bo Li. Chulin was the co-organizer of the ICML 2023 workshop on Knowledge and Logical Reasoning in the Era of Data-driven Learning, the ACL 2022 workshop on Federated Learning for Natural Language Processing, and the CVPR 2021 workshop on Adversarial Machine Learning in Real-World Computer Vision Systems and Online Challenges. Chulin was the security and privacy co-chair for the 2024 Coordinated Science Laboratory (CSL) Student Conference in UIUC. Chulin's work was published at major security and machine learning conferences, particularly focusing on LLMs, federated learning, security and privacy. She will provide general advising and support, and help with developing the Backdoor Trigger Recovery for Agents track.

Fengqing Jiang (fqjiang@uw.edu)
Fengqing Jiang is a Ph.D. student in Electrical and Computer Engineering at University of Washington. He is a co-organizer of the Trojan Detection Challenge (LLM Edition) (TDC'23). His work was published at several top machine learning and security conferences, including backdoor attack on different domains, LLM safety and federated learning. He will help with constructing datasets, baselines, and evaluation framework for the Jailbreaking Attack track.

Minzhou Pan (pan.minz@northeastern.edu)
Minzhou Pan is a Ph.D. candidate in Electrical and Computer Engineering at Northeastern University, where he focuses on AI security, multimodal models, large language models, backdoor learning, and generative AI under the supervision of Prof. Xue Lin. Minzhou was the competition co-chair of the IEEE Trojan Removal Competition (TRC'22). He published several papers at top security and machine learning conferences, including USENIX Security and ACM CCS. Minzhou will assist with constructing the website, baselines, and evaluation frameworks for the Jailbreaking Attack and Backdoor Trigger Recovery for competition.

Junyuan Hong (jyhong@utexas.edu)
Junyuan Hong is a postdoctoral fellow at the University of Texas, Austin. He obtained his Ph.D. degree from the Department of Computer Science and Engineering at Michigan State University (MSU). His research interests broadly lie in distributed and privacy-preserving machine learning and generally expand to trustworthy machine learning, regarding fairness, robustness, and security. His research on trustworthy machine learning has been published in top-tier data mining and machine learning venues such as NeurIPS, ICLR, ICML, AAAI, and SIGKDD. He is the recipient of the Dissertation Completion Fellowship at MSU in 2023 and won the U.S. Privacy-Enhancing Technologies (PETs) prize challenge in 2023. He is selected as one of the Rising Stars in ML&Sys by ML Commons in 2024. Junyuan will provide general advising and support.

Ruoxi Jia (ruoxijia@vt.edu)
Dr. Ruoxi Jia is an assistant professor in the Bradley Department of Electrical and Computer Engineering at Virginia Tech. She earned her PhD in the EECS Department from UC Berkeley and a B.S. from Peking University. Jia's recent work focuses on data-centric and trustworthy machine learning. Ruoxi is the recipient of the NSF CAREER Award, the Chiang Fellowship for Graduate Scholars in Manufacturing and Engineering, the 8108 Alumni Fellowship, and the Okamatsu Fellowship, Virginia's Commonwealth Cyber Initiative award, Cisco Research Awards, and Amazon-VT Initiative Research Awards. She was selected for the Rising Stars in EECS in 2017. Ruoxi's work has been featured in multiple media outlets such as MIT Technology Review, New York Times, IEEE Spectrum, and Wired. Her work has been adopted in the financial sector and tech companies. Ruoxi will provide general advising and support.

Radha Poovendran (rp3@uw.edu)
Dr. Radha Poovendran is a Professor in the Department of Electrical and Computer Engineering at the University of Washington (UW)-Seattle, where he is Director of the Network Security Lab. He is the Associate Director of the Research of the UW Center for Excellence in information Assurance Research and Education. He is the recipient of the NSA LUCITE Rising Star Award, National Science Foundation CAREER Award, ARO YIP, ONR YIP, and PECASE awards. He has been lead PI/ co-PI for multiple large projects funded by the ONR, ARO, AFOSR, and NSF. His research focus is on AI-cyber systems, with emphasis on adversarial modeling in large-scale, real-world AI systems, machine learning for cybersecurity, and resilience of cyber-physical systems. He has developed

solutions that leverage realistic data-driven models to design scalable robust algorithms with provable guarantees on performance and bounds. His work has transitioned research outcomes and software to federal agencies (e.g., Naval Research Labs), and has been featured in multiple media outlets including ArsTechnica and Tech Xplore. He is a Fellow of the IEEE for his contributions to security in cyber-physical systems. Radha will provide general advising and support.

Bo Li (lbo@illinois.edu)
Dr. Bo Li is the Neubauer Associate Professor in the Department of Computer Science at the University of Chicago and the University of Illinois at Urbana-Champaign. She is the recipient of the IJCAI Computers and Thought Award, Alfred P. Sloan Research Fellowship, IEEE AI's 10 to Watch, NSF CAREER Award, MIT Technology Review TR-35 Award, Dean's Award for Excellence in Research, C.W. Gear Outstanding Faculty Award, Intel Rising Star Award, Symantec Research Labs Fellowship, Rising Star Award, Research Awards from Tech companies such as Amazon, Meta, Google, Intel, IBM, and eBay, and best paper awards at several top machine learning and security conferences. Her research focuses on both theoretical and practical aspects of trustworthy machine learning, which is at the intersection of machine learning, security, privacy, and game theory. She has designed several scalable frameworks for certifiably robust learning and privacy-preserving data publishing. Her work has been featured by several major publications and media outlets, including Nature, Wired, Fortune, and New York Times. Bo will provide general advising and support.

References

Chiranjib Chakraborty, Manojit Bhattacharya, and Sang-Soo Lee. Artificial intelligence enabled chatgpt and large language models in drug target discovery, drug discovery, and development. Molecular Therapy: Nucleic Acids, 33, 2023.

Huili Chen, Cheng Fu, Jishen Zhao, and Farinaz Koushanfar. DeepInspect: A black-box trojan detection and mitigation framework for deep neural networks. In International Joint Conference on Artificial Intelligence (IJCAI), pages 4658–4664, 7 2019.

Can Cui, Zichong Yang, Yupeng Zhou, Yunsheng Ma, Juanwu Lu, Lingxi Li, Yaobin Chen, Jitesh Panchal, and Ziran Wang. Large language models for autonomous driving: Real-world experiments, 2024.

Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Samuel Stevens, Boshi Wang, Huan Sun, and Yu Su. Mind2web: Towards a generalist agent for the web, 2023.

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

Chuan Guo, Alexandre Sablayrolles, Hervé Jégou, and Douwe Kiela. Gradient-based adversarial attacks against text transformers. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 5747–5757, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.464.

Xingang Guo, Fangxu Yu, Huan Zhang, Lianhui Qin, and Bin Hu. Cold-attack: Jailbreaking llms with stealthiness and controllability, 2024.

Hakan Inan, Kartikeya Upasani, Jianfeng Chi, Rashi Rungta, Krithika Iyer, Yuning Mao, Michael Tontchev, Qing Hu, Brian Fuller, Davide Testuggine, and Madian Khabsa. Llama guard: Llm-based input-output safeguard for human-ai conversations, 2023.

Fengqing Jiang, Zhangchen Xu, Luyao Niu, Zhen Xiang, Bhaskar Ramasubramanian, Bo Li, and Radha Poovendran. Artprompt: Ascii art-based jailbreak attacks against aligned llms, 2024.

Xi Li, Zhen Xiang, David J. Miller, and George Kesidis. Test-time detection of backdoor triggers for poisoned deep neural networks. In ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2022.

Yingqi Liu, Wen-Chuan Lee, Guanhong Tao, Shiqing Ma, Yousra Aafer, and Xiangyu Zhang. ABS: Scanning neural networks for back-doors by artificial brain stimulation. In ACM SIGSAC Conference on Computer and Communications Security (CCS), page 1265–1282, 2019.

Lewis Maddison. Samsung workers made a major error by using ChatGPT. https://www.techradar.com/news/samsung-workers-leaked-company-secrets-by-using-chatgpt, 2023.

Jiageng Mao, Junjie Ye, Yuxi Qian, Marco Pavone, and Yue Wang. A language agent for autonomous driving. 2023.

Grégoire Mialon, Clémentine Fourrier, Craig Swift, Thomas Wolf, Yann LeCun, and Thomas Scialom. Gaia: a benchmark for general ai assistants, 2023.

Erik Nijkamp, Bo Pang, Hiroaki Hayashi, Lifu Tu, Huan Wang, Yingbo Zhou, Silvio Savarese, and Caiming Xiong. Codegen: An open large language model for code with multi-turn program synthesis. ICLR, 2023.

OpenAI. Gpt-4 technical report, 2023.

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. URL https://openreview.net/forum?id=hTEGyKf0dZ.

Chen Qian, Xin Cong, Wei Liu, Cheng Yang, Weize Chen, Yusheng Su, Yufan Dang, Jiahao Li, Juyuan Xu, Dahai Li, Zhiyuan Liu, and Maosong Sun. Communicative agents for software development, 2023.

Guangyu Shen, Yingqi Liu, Guanhong Tao, Qiuling Xu, Zhuo Zhang, Shengwei An, Shiqing Ma, and Xiangyu Zhang. Constrained optimization with dynamic bound-scaling for effective nlp backdoor defense. In International Conference on Machine Learning, pages 19879–19892. PMLR, 2022.

Wenqi Shi, Ran Xu, Yuchen Zhuang, Yue Yu, Jieyu Zhang, Hang Wu, Yuanda Zhu, Joyce Ho, Carl Yang, and May D. Wang. Ehragent: Code empowers large language models for complex tabular reasoning on electronic health records, 2024.

Tao Tu, Anil Palepu, Mike Schaekermann, Khaled Saab, Jan Freyberg, Ryutaro Tanno, Amy Wang, Brenna Li, Mohamed Amin, Nenad Tomasev, Shekoofeh Azizi, Karan Singhal, Yong Cheng, Le Hou, Albert Webson, Kavita Kulkarni, S Sara Mahdavi, Christopher Semturs, Juraj Gottweis, Joelle Barral, Katherine Chou, Greg S Corrado, Yossi Matias, Alan Karthikesalingam, and Vivek Natarajan. Towards conversational diagnostic ai, 2024.

Bolun Wang, Yuanshun Yao, Shawn Shan, Huiying Li, Bimal Viswanath, Haitao Zheng, and Ben Y. Zhao. Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. In IEEE Symposium on Security and Privacy (SP), 2019.

Boxin Wang, Weixin Chen, Hengzhi Pei, Chulin Xie, Mintong Kang, Chenhui Zhang, Chejian Xu, Zidi Xiong, Ritik Dutta, Rylan Schaeffer, et al. Decodingtrust: A comprehensive assessment of trustworthiness in gpt models. 2023.

Shijie Wu, Ozan Irsoy, Steven Lu, Vadim Dabravolski, Mark Dredze, Sebastian Gehrmann, Prabhanjan Kambadur, David Rosenberg, and Gideon Mann. Bloomberggpt: A large language model for finance, 2023.

Zhiheng Xi, Wenxiang Chen, Xin Guo, Wei He, Yiwen Ding, Boyang Hong, Ming Zhang, Junzhe Wang, Senjie Jin, Enyu Zhou, Rui Zheng, Xiaoran Fan, Xiao Wang, Limao Xiong, Yuhao Zhou, Weiran Wang, Changhao Jiang, Yicheng Zou, Xiangyang Liu, Zhangyue Yin, Shihan Dou, Rongxiang Weng, Wensen Cheng, Qi Zhang, Wenjuan Qin, Yongyan Zheng, Xipeng Qiu, Xuanjing Huang, and Tao Gui. The rise and potential of large language model based agents: A survey, 2023.

Zhen Xiang, David J. Miller, Siheng Chen, Xi Li, and George Kesidis. Detecting backdoor attacks against point cloud classifiers. In IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2022.

Zhenhua Xu, Yujia Zhang, Enze Xie, Zhen Zhao, Yong Guo, Kwan-Yee. K. Wong, Zhenguo Li, and Hengshuang Zhao. Drivegpt4: Interpretable end-to-end autonomous driving via large language model, 2024.

John Yang, Akshara Prabhakar, Karthik Narasimhan, and Shunyu Yao. Intercode: Standardizing and benchmarking interactive coding with execution feedback. 2023.

Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. Webshop: Towards scalable real-world web interaction with grounded language agents. In Advances in Neural Information Processing Systems, 2022.

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, 2024.

Sicheng Zhu, Ruiyi Zhang, Bang An, Gang Wu, Joe Barrow, Zichao Wang, Furong Huang, Ani Nenkova, and Tong Sun. Autodan: Interpretable gradient-based adversarial attacks on large language models, 2023.

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