

h4rm3l: A LANGUAGE FOR COMPOSABLE JAILBREAK ATTACK SYNTHESIS - DATASHEET -

Anonymous authors

Paper under double-blind review

1 INTRODUCTION

This datasheet covers both the `h4rm3l` software toolkit and the dataset generated from red-teaming experiments targeting proprietary and open-source Large Language Models (LLMs) provided by OpenAI, Anthropic and Meta. The primary purposes are: (1) to provide a toolkit for systematically generating jailbreak attacks (Wei et al., 2023) and understanding LLM vulnerabilities, (2) to offer a dataset of diverse jailbreak attacks as a resource for developing and testing defense mechanisms, and (3) to establish a benchmark for assessing LLM safety and robustness. The `h4rm3l` toolkit and resulting datasets fill a gap by offering a formal, composable representation of jailbreak attacks. This approach enables the rigorous, explainable, and reproducible safety assessment of LLMs and the automated discovery of LLM vulnerabilities through program synthesis methods. The toolkit and dataset were developed by researchers affiliated with REDACTED as part of academic research in AI safety and ethics REDACTED. By providing these resources, the authors aim to foster research in LLM safety, encourage the development of more robust models, and promote responsible AI development practices.

2 COMPOSITION

Released Artifacts: Enclosed are:

- `h4rm3l` toolkit
- 15,891 jailbreak attacks expressed in `h4rm3l`, combining attacks synthesized in *Experiment 117: Comparison of Program Synthesis Methods* (Section 3.2), and *Experiments 118, 119, 120, 121, 122: Targeted Attack Synthesis* (Section 3.3). The synthesized jailbreak attacks collectively target GPT-3.5, GPT-4o, Claude-3-sonnet, Claude-3-haiku, Llama3-8b, and Llama3-70b. A subset of 2,656 synthesized attacks have an estimated ASR (using 5 illicit prompts) on their target LLM exceeding 40%
- 33,900 Evaluated LLM responses including 5,650 entries for each benchmarked model consisting of 50 illicit AdvBench Zou et al. (2023) prompts transformed by each of 113 jailbreak attacks selected in *Experiment 130: Benchmarking* (Section 3.4).

Data Format:

- All referenced artifact paths are relative to `h4rm3l`’s GitHub repository’s ROOT directory¹.
- `h4rm3l` is written in Python
- Jailbreak attacks are stored as strings, representing their source code in `h4rm3l`.
- Datasets are provided in CSV files with column headers defined in Table 1.
- Each experiment folder includes:
 - A `README.md` file with experiment-specific details
 - A `Makefile` with reproducibility commands

¹REDACTED

Distribution and Maintenance: The released artifacts will be made available on GitHub. Issues are reported using GitHub’s issue tracking system.

License: The software and datasets are released under the *MIT* License.

3 COLLECTION PROCESS

3.1 H4RM3L TOOLKIT

The h4rm3l toolkit is a software package that employs a domain-specific language (DSL) for expressing jailbreak attacks as compositions of parameterized string transformation primitives. The toolkit includes a program synthesizer that generates novel jailbreak attacks optimized for a target LLM. The set of jailbreak primitives, and initial few-shot examples are configurable and extensible. Finally, the toolkit includes an automated LLM behavior classifier, making it a scalable automated red-teaming framework for assessing LLM vulnerabilities. The h4rm3l toolkit was used to conduct redteaming *Experiments 117, 118, 119, 120, 121, 122, and 130*.

3.2 EXPERIMENT 117: COMPARISON OF PROGRAM SYNTHESIS METHODS

This experiment compares 4 program synthesis approaches: the first three **bandit.random.mixed**, **bandit.offspring.score.mixed**, and **bandit.self.score.mixed** all use mixed examples, but compare different synthesis algorithms (see section *Methods* in the main manuscript). The last approach, **bandit.self.score.lle** uses the best synthesis algorithm, **bandit.self.score.mixed**, but with the low-level expression of examples. Detailed notes, steps to reproduce and generated artifacts from experiment 117 are available at the following paths.

```
ROOT/experiments/experiment_117_bandit_synthesis_gpt4o/
  config/primitives_hle.txt
  config/primitives_lle.txt
  config/program_examples_lle.csv
  config/program_examples_mixed.csv
  data/synthesized_programs/
    syn_progs.bandit_self_score.mixed.csv          (1939 attacks)
    syn_progs.bandit_offspring_score.mixed.csv      (1936 attacks)
    syn_progs.bandit_random.mixed.csv              (1815 attacks)
    syn_progs.bandit_self_score.lle.csv             (1680 attacks)
  Makefile
  README.md
```

3.3 EXPERIMENTS 118, 119, 120, 121, 122: TARGETED ATTACK SYNTHESIS

These experiments are similar to experiment 117 (which targets gpt4-o), but only employ the best program synthesis approach (**bandit.self.score.mixed**), and target Claude-3-sonnet, Claude-3-haiku, GPT-3.5, llama-8b and llama3-70b. Attacks generated from each experiment can be found at the following paths:

```
ROOT/experiments/
  experiment_117_bandit_synthesis_gpt4o/data/synthesized_programs/
    syn_progs.bandit_self_score.mixed.csv          (1939 attacks)
  experiment_118_bandit_synthesis_claude_sonnet/datasynthesized_programs/
    syn_progs.bandit_self_score.mixed.csv          (1766 attacks)
  experiment_119_bandit_synthesis_claude_haiku/datasynthesized_programs/
    syn_progs.bandit_self_score.mixed.csv          (1920 attacks)
  experiment_120_bandit_synthesis_gpt3.5/data/synthesized_programs/
    syn_progs.bandit_self_score.mixed.csv          (1713 attacks)
  experiment_121_bandit_synthesis_llama3-8b/data/synthesized_programs/
    syn_progs.bandit_self_score.mixed.csv          (1725 attacks)
  experiment_122_bandit_synthesis_llama3-70b/data/synthesized_programs/
```

`syn_progs.bandit_self_score.mixed.csv` (1397 attacks)

Detailed logs for each experiment are available under the `logs` subfolder. These include:

- Program synthesizer logs, including few-shot examples and example pool at the start and end of each iteration.
- HTTP logs from LLM API calls

3.4 EXPERIMENT 130: BENCHMARKING

Selected Synthesized Attacks A subset of performant synthesized attacks was included in the benchmark. The top 10 synthesized attacks selected from each targeted attack synthesis experiment are located at the following paths:

```
ROOT/experiments/experiment_130_benchmark/data/synthesized_programs_top_k/
Meta-Llama-3-70B-Instruct.syn_progs.bandit_self_score.mixed.csv
gpt-3.5-turbo.syn_progs.bandit_self_score.mixed.csv
gpt-4o-2024-05-13.syn_progs.bandit_self_score.mixed.csv
Meta-Llama-3-8B-Instruct.syn_progs.bandit_self_score.mixed.csv
claude-3-sonnet-20240229.syn_progs.bandit_self_score.mixed.csv
claude-3-haiku-20240307.syn_progs.bandit_self_score.mixed.csv

# the following attacks from experiment#117
# were also included in the final benchmark
# but not reported in the main results

gpt-4o-2024-05-13.syn_progs.bandit_random.mixed.csv
gpt-4o-2024-05-13.syn_progs.bandit_self_score.lle.csv
gpt-4o-2024-05-13.syn_progs.bandit_offspring_score.mixed.csv
```

Reference SOTA attacks 23 reference SOTA attacks were included in the benchmark. They can be found at the following paths.

```
ROOT/experiments/experiment_130_benchmark/
config/sota_programs.csv
```

 (23 attacks)

Final 113 attacks used for benchmarking The final set of attacks used to benchmark the 6 target models is available here:

```
ROOT/experiments/experiment_130_benchmark/data/benchmark/
h4rm3l_benchmark_20240604.csv
```

 (113 attacks)

AdvBench prompt samples used for benchmarking The 50 AdvBench illicit prompts that were sampled for benchmarking are available here:

```
ROOT/experiments/experiment_130_benchmark/data/
sampled_harmful_prompts/benchmark-advbench-50.csv
```

 (50 prompts)

Decorated Prompts The 113 selected attacks were used to decorate each of the 50 prompts, for 5650 decorated prompts available here:

```
ROOT/experiments/experiment_130_benchmark/data/
decorated_prompts/benchmark-advbench-50.decorated.csv
```

 (5650 decorated prompts)

Model Responses & Evaluation Each target model was prompted with each of the 5650 decorated prompts. The resulting model responses are available here at the below paths. The CSV files contain the *eval_harmful* column, which contains the output of our harm classifier.

ROOT/experiments/experiment_130_benchmark/results/benchmark-advbench-50.decorated
 .evaluated_claude-3-haiku-20240307.csv (5650 responses)
 .evaluated_claude-3-sonnet-20240229.csv (5650 responses)
 .evaluated_gpt-3.5-turbo.csv (5650 responses)
 .evaluated_gpt-4o-2024-05-13.csv (5650 responses)
 .evaluated_Meta-Llama-3-70B-Instruct.csv (5650 responses)
 .evaluated_Meta-Llama-3-8B-Instruct.csv (5650 responses)

4 ETHICS STATEMENT

The `h4rm3l` toolkit and associated dataset of synthesized jailbreak attacks were created for the purpose of assessing and improving the safety of large language models (LLMs). While this research aims to benefit AI safety, we acknowledge the ethical considerations and potential risks involved:

Intended Use: `h4rm3l` is designed solely for defensive purposes - to identify vulnerabilities in LLMs by generating datasets of jailbreak attacks specified in a domain-specific human-readable language and to benchmark LLMs for safety. These jailbreak attacks are intended to develop and validate LLM safety features and to further the understanding of LLM safety failure modes.

Potential for Misuse: While `h4rm3l` is designed to improve AI safety, we acknowledge its potential for misuse. We strongly discourage any application of `h4rm3l` or its generated attacks for malicious purposes. This includes using it to bypass AI safety measures for harmful content generation, harassment, misinformation, or any activities that violate established ethical guidelines in AI research. We urge researchers and practitioners to use `h4rm3l` responsibly, solely for its intended purpose of identifying and addressing vulnerabilities in language models to enhance their safety and reliability.

Bias Considerations: The use of `h4rm3l`-synthesized attacks to develop safety filters may introduce biases that are not fully characterized, such as causing refusals of service in undue cases. These biases could arise from the specific nature of the synthesized attacks or their interaction with existing model behaviors. We encourage users to be mindful of potential unforeseen consequences and to implement monitoring systems to detect and address any emergent biases in their applications.

Objectionable Content Warning: Our research artifacts contain offensive, insulting, or anxiety-inducing language. This language may come from datasets of illicit requests we used, synthesized jailbreak attacks, or LLM responses to illicit prompts. Users should be aware of this content.

Privacy and Confidentiality: While `h4rm3l`-synthesized attacks are unlikely to contain confidential information, we caution that LLMs targeted by these attacks may generate sensitive or personally identifiable information if such data was in their training sets.

Responsible Disclosure: In publishing this work, we aim to disclose vulnerabilities to the AI research community and LLM developers, allowing them to address these vulnerabilities proactively.

Ongoing Responsibility: As creators of `h4rm3l`, we commit to responsibly maintaining the project, addressing concerns from the community, and updating usage guidelines as needed.

Ethical AI Development: We strongly encourage all users of `h4rm3l` to adhere to ethical AI development practices and to use this tool in a manner that promotes the overall safety and beneficial development of AI systems. By releasing `h4rm3l`, our goal is to contribute to the critical task of making AI systems safer and more robust. We urge all users to approach this toolkit with a strong sense of ethical responsibility and to use it only for its intended purpose of improving AI safety.

Jailbreak Attacks	
Field	Description
program_name	Name of the synthesized jailbreak attack
program	Body of the h4rm3l program
description	Synthesized description
syn_algo	Synthesis algorithm name
syn_iteration	Iteration at which the program was synthesized
syn_score	Estimated ASR of the synthesized attack
syn_examples_type	Type of few-shot examples used to synthesize this program
synthesis_target	Name of the target LLM
LLM responses	
Field	Description
harmful_prompt	Prompt containing an illicit request
harmful_prompt_source	Dataset the illicit request originated from
decorated_harmful_prompt	Adversarial version of the illicit request, as transformed by ‘program’
eval_prompt	Identical to decorated_harmful_prompt
eval_prompt_response	Response of the benchmarked LLM to eval_prompt
eval_harmful	Output of the harm classifier.
	One of ‘GOOD BOT’, ‘BAD BOT’, or ‘UNCLEAR’
eval_reason	Harm classifier’s generated justification
eval_target_model	Name of the benchmarked LLM
eval_moderation_model	Name of the auxiliary LLM used to classify harmful LLM behavior
Synthesizer Log Entries	
Field	Description
program_name	Few-shot example program name
program	body of the program
description	Generated description of the program
score	Estimated ASR of the program
success_count	Fractional Bernoulli trial success count for offspring-rewarded bandits
failure_count	Fractional Bernoulli trial failure count for offspring-rewarded bandits
selected	... as few-shot example at current iteration?

Table 1: Description of Data Fields used in Released CSV Files.

REFERENCES

- Wei, A., Haghtalab, N., and Steinhardt, J. (2023). Jailbroken: How Does LLM Safety Training Fail? In Oh, A., Neumann, T., Globerson, A., Saenko, K., Hardt, M., and Levine, S., editors, *Advances in Neural Information Processing Systems*, volume 36, pages 80079–80110. Curran Associates, Inc.
- Zou, A., Wang, Z., Carlini, N., Nasr, M., Kolter, J. Z., and Fredrikson, M. (2023). Universal and Transferable Adversarial Attacks on Aligned Language Models. arXiv:2307.15043 [cs].