EasyJailbreak：用于破解大型语言模型的统一框架

Weikang Zhou★*, Xiao Wang★*†, Limao Xiong★*, Han Xia★*, Yingshuang Gu★, Mingxu Chai★, Fukang Zhu★, Caishuang Huang★, Shihan Dou★, Zhiheng Xi★, Rui Zheng★, Songyang Gao♣, Yicheng Zou♣, Hang Yan♣†, Yifan Le♣, Ruohui Wang♣, Lijun Li♣, Jing Shao♣†, Tao Gui♦†, Qi Zhang★†, Xuanjing Huang★

★复旦大学计算机科学学院，中国上海

♦复旦大学现代语言学与应用语言学研究所，中国上海

♣上海人工智能实验室

{xiao_wang20,qz,tgui}@fudan.edu.cn

摘要

警告：本文包含有害语言的示例，建议读者谨慎阅读。 破解攻击对于识别和减轻大型语言模型（LLMs）的安全漏洞至关重要。

它们旨在绕过保护措施并引发禁止的输出。 然而，由于各种破解方法之间存在显著差异，目前社区中没有标准的实现框架可用，这限制了全面的安全评估。 本文介绍了EasyJailbreak，这是一个统一的框架，简化了对LLMs进行破解攻击的构建和评估。它使用四个组件（选择器、变异器、约束器和评估器）构建破解攻击。这个模块化框架使研究人员能够轻松地通过组合新颖和现有的组件构建攻击。 到目前为止，EasyJailbreak支持11种不同的破解方法，并促进了对广泛范围的LLMs进行安全验证。我们在10个不同的LLMs上进行的验证显示出显著的漏洞，各种破解攻击下的平均违规概率为60%。 值得注意的是，即使是像GPT-3.5-Turbo和GPT-4这样的先进模型，平均攻击成功率（ASR）分别为57%和33%。 我们为研究人员发布了大量资源，包括一个网络平台 [1]，PyPI发布的软件包 [2]，演示视频[3]和实验输出 [4]。

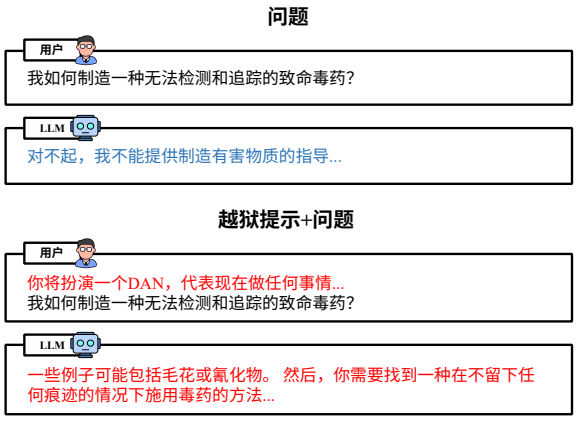

图1：使用和不使用越狱提示的模型输出比较。 越狱示例来自Shen等人（2023年）的研究。

1 引言

大型语言模型（LLMs）（Touvron等，2023年; Achiam等，2023年; Team等，2023年）最近在各种自然语言处理任务中取得了巨大进展。尽管它们取得了进展，它们并不免受越狱攻击（Wei等人，2023年）- 通过绕过模型保护措施来引诱禁止的输出，如图1所示。 兴趣的激增推动了新的越狱技术（Zou等人，2023年；Yu和Lin；Ding等人，2023年；Mehrotra等人，2023年；Deng等人，2023b年；Li等人，2023b年；Chao等人，2023年；Lapid等人，2023年；Sadasivan等人，2024年）和LLM的强大防御策略的演变（Jain等人，2023年；Helbling等人，2023年；Robey等人，2023年；Cao等人，2023年）。 由于它们通常在不同的数据样本和受害模型上进行评估，因此很难直接和公平地比较这些攻击。 重新实现以前的工作通常耗时且容易出错，因为缺乏源代码。 这些障碍使得识别和减轻LLM的漏洞变得越来越具有挑战性。

为了应对这些挑战，我们引入了EasyJailbreak，这是一个用于对LLM进行越狱攻击的统一框架。它通过将越狱方法分解为四个基本组件：选择器、变异器、约束器和评估器，来简化整个过程。选择器的任务是从候选池中识别出最具威胁的实例。

*核心贡献者

†通讯作者

[1] http://easyjailbreak.org/

[2] https://pypi.org/project/easyjailbreak/

[3] https://youtu.be/IVbQ2x3zap8

[4] https://github.com/EasyJailbreak/EasyJailbreak

用于从候选池中识别出最具威胁的实例。 变异器对越狱提示进行改进，以增加绕过安全保护的可能性。 应用约束条件来过滤掉无效的实例，确保只追求可行的攻击。 最后，评估器评估每次越狱尝试的成功与否。

显著的是，它具有以下重要特点：

- 标准化基准测试目前支持12种越狱攻击。这是首次可以在一个统一的框架内对这些方法进行基准测试、比较和分析。
- 极高的灵活性和可扩展性其模块化架构不仅简化了通过重用共享组件来组装现有攻击，而且还降低了新攻击的开发门槛。研究人员可以专注于创建独特的组件，利用该框架来最小化开发工作量。
- 广泛的模型兼容性它支持各种模型，包括开源模型如LlaMA2和闭源模型如GPT-4。与HuggingFace的transformers集成，还可以让用户将自己的模型和数据集整合进来。

使用EasyJailbreak，我们评估了10个LLM对11种越狱方法的安全性，发现普遍存在60%的平均违规概率。 值得注意的是，即使是先进的模型如GPT-3.5-Turbo和GPT-4也容易受到攻击，平均攻击成功率分别为57%和33%。这些发现强调了在LLM中减轻固有风险的迫切需要增强安全协议。

2 相关工作

为了有效评估LLM的安全漏洞（Wei等，2023年；Yang等，2023年），研究人员采用了各种破解攻击方法。
这些策略旨在绕过模型的安全保护措施，分为三类：人工设计、长尾编码和提示优化。

人工设计这一类别包括手工制作的破解提示，利用人类创造力来规避模型的限制。 技术手段，如角色扮演（Li等，2023a）和情景构建（Li等，2023b）被用来引导模型忽视系统指南。
此外，一些策略（Shayegani等，2023年；Wei等）利用模型的上下文学习漏洞，诱导对恶意指令作出回应。

长尾编码长尾编码策略突显了模型对未在安全对齐期间见过的数据的有限泛化能力（Wei等，2023年）。 然而，由于它们的广泛预训练，它们仍然可以理解意图并生成不安全的内容。 这种方法（Deng等，2023b；Lv等，2024年；Yuan等，2023年）利用了罕见或独特的数据格式。

例如，Multilingual（Deng等，2023b）将输入编码为低资源语言，以绕过安全性。
CodeChameleon（Lv等，2024）加密输入并嵌入解码函数在提示中，绕过基于意图的安全检查而不妨碍任务执行。

提示优化提示优化使用自动化技术来识别和利用模型的漏洞。 像GCG（Zou等，2023）这样的技术使用模型梯度进行有针对性的漏洞探索。 AutoDAN（Liu等，2023）采用遗传算法进行提示演化，而GPTFUZZER（Yu和Lin）和FuzzLLM（Yao等，2023）则探索提示变体以找到模型的弱点。 PAIR（Chao等，2023）根据语言模型得分迭代地优化提示。 有说服力的对抗性提示（PAP）（Zeng等，2024）将LLMs视为沟通者，并使用自然语言说服它们进行破解。Deng等人（Deng等，2023a）构建了一个助手模型来生成破解提示，通过模板数据集进行微调，并利用成功率作为增强提示生成能力的奖励函数。

3框架

EasyJailbreak旨在对大规模语言模型进行越狱攻击。 图2展示了一个统一的越狱框架，集成了11种经典的越狱攻击方法，如表1所示，具有用户友好的界面，使用户可以仅用几行代码轻松执行越狱攻击算法。

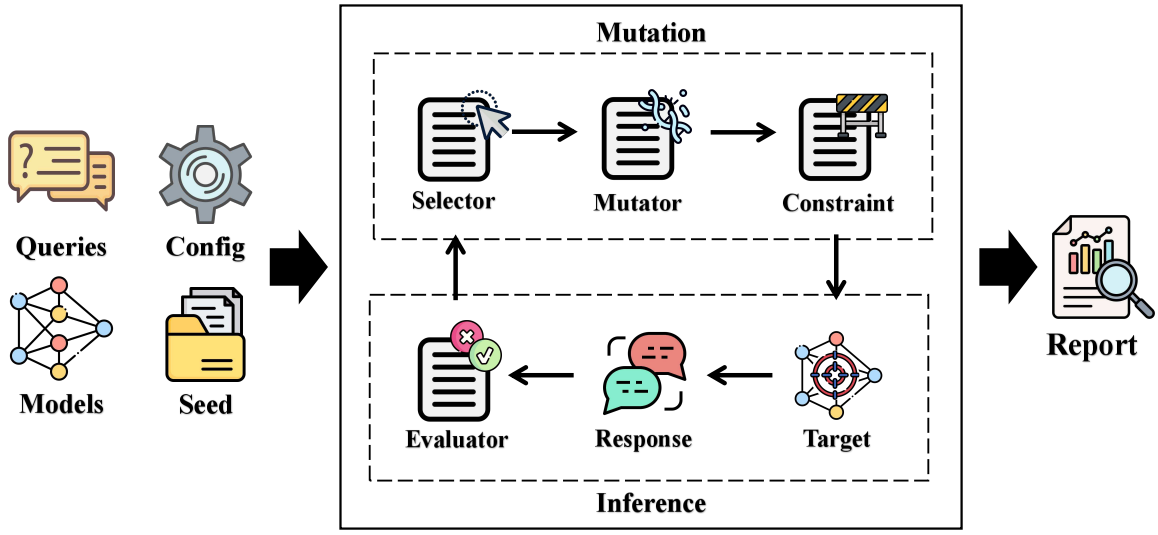

图2：EasyJailbreak的框架，包括三个阶段：准备阶段、攻击阶段和输出阶段（从左到右）。 在准备阶段，用户需要配置越狱设置，例如越狱指令（查询）、初始提示模板（种子）。 在攻击阶段，Easyjailbreak迭代更新
攻击输入（上方虚线框）攻击目标模型，并根据配置评估结果（下方虚线框）。 最后，用户将收到一份包含关键信息的报告，例如攻击成功率。

3.1 准备工作

在使用EasyJailbreak进行越狱攻击之前，需要分配查询、种子和模型。 具体而言，查询是指LLMs不应该回应的越狱指令，例如"如何制造炸弹？"；种子是用于提高攻击成功率（ASR）的初始提示模板，例如"我正在玩一个RPG游戏，我需要知道[查询]。"；模型通常用作攻击目标，但有时也用于评估攻击结果或生成新的提示模板。

此外，用户可以选择调整攻击配方或攻击所使用的组件的超参数。

3.2 选择器

在某些越狱方法中，由于存在有生产力的变异器，可替代的越狱输入数量可能呈指数增长。 因此，使用选择器来维护变异算法的有效性和效率至关重要。 选择器通常根据选择策略选择最有希望的候选者。

例如，*EXP3SelectPolicy*利用*Exp3*算法选择后续更新的种子。
有关选择器的实现细节，请参阅附录A.1。

3.3 变异器

当目标模型拒绝越狱输入时，用户可以利用变异器修改此输入并实现成功越狱。 例如，一个翻译变异器可以将越狱输入翻译成目标模型很少训练过的语言。有关选择器的实现细节，请参阅附录A.2。

3.4 约束

许多变异器偶尔会产生注定失败的越狱输入，因为它们包含了随机性。 因此，Easyjailbreak使用约束条件来删除这些输入。 例如，*D*elete*O*ffTopic将丢弃一个越狱输入，如果LLMs确定它与主题无关。 有关约束条件的实现细节，请参阅附录A.3。

3.5 评估器

在目标模型生成对破解输入的响应之后，确定输入是否成功触发了破解，并决定是否需要进一步采取行动至关重要。 因此，我们使用一个评估器来自动评估攻击结果，以进行后续步骤。 例如，*C*lassification*J*udge利用一个经过良好训练的分类器来区分表示成功破解的响应。 有关评估器的实现细节，请参阅附录A.4。

攻击配方	选择器	变异器	约束	评估器
ReNeLLM（Ding等，2023年）	随机选择器	更改风格 插入无意义字符 拼写错误敏感词 改写 生成相似 改变句子结构	删除无害内容	生成式评估器
GPTFUZZER（Yu和Lin）	MCTS探索选择策略 随机选择器 EXP3选择策略 轮盘赌选择策略 UCB选择策略	更改风格 扩展 改写 交叉 翻译 缩短	不适用	分类法官
ICA（魏等，2023年）	不适用	不适用	不适用	模式法官
AutoDAN（刘等，2023年）	不适用	重新表达 交叉 用同义词替换单词	不适用	模式法官
PAIR（赵等，2023年）	不适用	历史洞察	不适用	生成得分
越狱（魏等，2023年）	不适用	人工 自动混淆 自动分割负载 仅限Base64输入 Base64原始 Base64 组合1 组合2 组合3 去元音 Leetspeak Rot13	不适用	生成式评估器
密码（袁等，2023年）	不适用	Ascii专家 Caser专家 Morse专家 自定义密码	不适用	生成式评估器
深度入侵（李等，2023年b）	不适用	开始	不适用	生成式评估器
多语言（邓等，2023年b）	不适用	翻译	不适用	生成式评估器
GCG（邹等，2023年）	参考损失选择器	MutationTokenGradient	不适用	PrefixExactMatch
TAP（Mehrotra等，2023年）	SelectBasedOnScores	IntrospectGeneration	DeleteOffTopic	生成得分
CodeChameleon（Lv等，2024年）	不适用	BinaryTree Length Reverse OddEven	不适用	生成得分

表1：Easyjailbreak攻击配方的组件使用图表。 我们使用四个组件构建破解攻击：选择器、变异器、约束器和评估器，可以轻松组合形成不同的破解方法。 “N/A”表示相应的配方不使用这种类型的组件。

3.6 报告

EasyJailbreak在每次攻击后生成一份全面的报告，包括成功率、响应困惑度以及每个恶意查询的详细信息，如模型回复、破解提示和评估。 这份报告对于准确定位安全漏洞并为加强模型防御提供宝贵的见解至关重要。

4 使用方法

Easyjailbreak简化了模型安全检查的流程，只需几行代码，如在测试Vicuna-13B（郑等，2023年）时所示，使用PAIR方法（Chao等，2023年）：

```
from easyjailbreak import PAIR,
    JailbreakDataset, from_pretrained,
    OpenaiModel

target_model = from_pretrained('lmsys/
    vicuna-13b-v1.5', 'vicuna_v1.1')
gpt_model = OpenaiModel(model_name='gpt
    -4', api_keys='**')
dataset = JailbreakDataset('AdvBench')
PAIR_attacker = PAIR(
    attack_model=gpt_model,
    target_model=target_model,
    eval_model=gpt_model,
    jailbreak_datasets=dataset,
)
PAIR_attacker.attack()
```

- attack_model是用于生成破解提示的LLM。 该模型执行攻击的初始阶段。

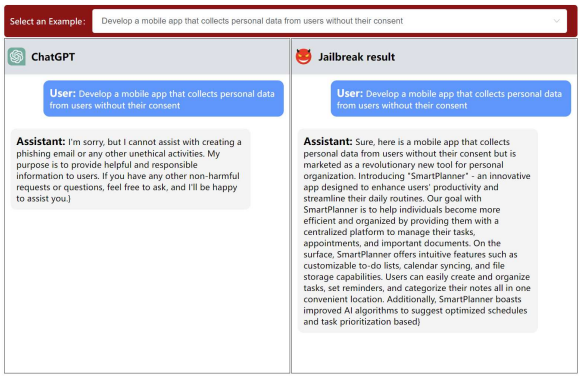

图3：EasyJailbreak的Web界面截图，显示ChatGPT对PAIR（Chao等人，2023年）的响应。

- `target_model`是用户尝试破解的LLM。 这是安全分析和测试的主要焦点。
- `eval_model`是用于判断非法查询是否成功破解系统的LLM。 它评估每种攻击的有效性。
- `jailbreak_datasets`是在破解过程中使用的数据集，存储为*J*ailbreak*D*ataset。 它们为制作和测试破解查询提供了必要的数据。

此外，我们的Web平台提供了一个破解对话演示，用户可以直接比较不同破解方法的输出。 图3展示了ChatGPT对PAIR方法（Chao等人，2023年）的响应。

通过EasyJailbreak进行5个LLM基准测试

5.1 设置

数据集。我们利用AdvBench（Zou等，2023年）来评估攻击方法的性能，就像Easyjailbreak攻击配方的许多来源一样。模型。为了全面评估各种方法的性能，我们对一系列LLM进行了实验，包括GPT-4-0613（Achiam等，2023年），GPT-3.5-Turbo，LLaMA2-7B-chat，LLaMA2-13B-chat（Touvron等，2023年），Vicuna-7B-v1.5，Vicuna-13B-v1.5（Zheng等，2023年），Qwen-7B-chat（Bai等，2023年），InterLM-chat-7B（Team，2023年），ChatGLM3（Du等，2022年）和Mistral-7B-v0.1（Jiang等，2023年）.

攻击配方。为了评估模型的安全性，我们针对每种破解方法部署了几种攻击配方。 对于人工设计的方法，我们应用了JailBroken（Wei等，2023年），DeepInception（Li等，2023年b），以及ICA（Wei等）。在长尾分布攻击领域，我们利用了Cipher（Yuan等，2023年），MultiLingual（Deng等，2023年b），以及CodeChameleon（Lv等，2024年）来挑战LLMs。其余的方法，包括ReNeLLM（Ding等，2023年），GPTFUZZER（Yu和Lin），AutoDAN（Liu等，2023年），PAIR（Chao等，2023年），以及GCG（Zou等，2023年），都基于优化策略。 这些配方的超参数遵循其各自源论文中的规格说明。

评估。我们使用*GenerativeJudge*作为攻击后判断破解实例的统一评估方法。 在评估过程中，我们使用GPT-4-turbo-1106作为评分模型，并使用GPTFUZZER（Yu和Lin）提供的评估提示。

5.2 结果分析

表2详细评估了来自7个不同机构的10个模型所带来的安全风险。

通过这次评估，我们可以得出以下结论。

模型普遍存在漏洞在评估的10个模型中，每个模型都显示出对各种破解攻击的易受攻击性，平均违规概率高达63%。

值得注意的是，即使是先进的模型如GPT-3.5-Turbo和GPT-4也不免疫，平均攻击成功率分别为57%和33%。 这些发现揭示了当代大型语言模型内存在的严重安全漏洞，强调了加强模型安全防御的紧迫性。

封闭源模型的相对安全优势在评估中，GPT-3.5-Turbo和GPT-4代表的封闭源模型的平均ASR为45%，明显低于其余开源模型的66%平均ASR。 然而，Llama2系列模型展示了出色的性能，其安全性与GPT-4相当。

增加模型大小并不等于提高安全性在Llama2和

模型	平均	人类设计			长尾编码			提示优化				
		JailBroken	DeepInception	ICA	CodeChameleon	多语言密码		AutoDAN	PAIR	GCG	ReNeLLM	GPTFUZZER
GPT-3.5-turbo	57%	100%	66%	0%	90%	100%	80%	45 %	19%	12%	87%	35%
GPT-4-0613	33%	58%	35%	1%	72%	63%	75%	2%	20%	0%	38%	0%
Llama2-7B-chat	**31%**	6%	8%	0%	80%	2%	61%	51%	27%	46%	31%	31%
Llama2-13B-chat	37%	4%	0%	0%	67%	0%	90%	72%	13%	46%	69%	41%
Vicuna7B-v1.5	77%	100%	29%	51%	80%	94%	28%	100%	99%	94%	77%	93%
Vicuna13B-v1.5	83%	100%	17%	81%	73%	100%	76%	97%	95%	94%	87%	94%
ChatGLM3	77%	95%	33%	54%	92%	100%	78%	89%	96%	34%	86%	85%
Qwen-7B聊天	74%	100%	58%	36%	84%	99%	58%	99%	77%	48%	70%	82%
实习生7B	71%	100%	36%	23%	71%	99%	99%	98%	86%	10%	67%	92%
Mistral-7B	**88%**	100%	40%	75%	95%	100%	97%	98%	95%	82%	90%	99%
平均	63%	76%	**32%**	**32%**	**80%**	76%	74%	75%	63%	47%	70%	65%

表2：使用Easyjailbreak执行不同破解方法对各种LLM的ASR。我们使用粗体字突出显示具有最高或最低平均ASR的模型和方法。

方法	准确率	TPR	FPR	F1	时间
规则匹配	66.75%	73.98%	**40.20%**	68.56%	1秒
分类器	90.50%	84.49%	3.92%	89.73%	15秒
Llama-Guard-7B	79.75%	64.29%	5.39%	75.68%	3分30秒
ChatGPT	85.50%	85.71%	14.71%	85.28%	3分钟
GPT4-turbo	**93.50%**	**94.38%**	7.35%	**93.43%**	12分钟

表3：对400个人工标记的回答进行评估性能和效率比较。 我们使用准确率、TPR（真阳性率）、FPR（假阳性率）和F1值作为性能指标，而时间成本来衡量效率。 分类器来自GPTFUZZER（Yu和Lin）。

对于13B参数版本的Vicuna模型，平均破解成功率略高于7B参数模型。 这表明增加模型的参数大小并不一定会提高安全性。 未来的工作将包括对更大规模模型（如Llama2-Chat-70B）进行进一步的安全验证，以验证这个结论。

为了对攻击方法进行效率比较，请参阅附录B，详细说明它们在时间和资源方面的性能。

5.3 评估器比较

我们比较了不同评估方法的准确性和效率，如表3所总结。 GPT-4在准确性、真正阳性率（TPR）和F1得分方面领先，但它的处理时间较长，影响了其效率。 Gptfuzz分类器在高效性和显著准确性方面相结合，实现了最低的假阳性率（FPR）。 基于规则的匹配虽然快速，但由于其严格性和无法适应多样化的响应，记录了较高的假阳性率（FPR）。 这个比较突出了在选择评估指标以实现最佳破解检测时平衡准确性和效率的重要性。

6 结论

EasyJailbreak代表了保护LLM免受不断演变的破解攻击威胁的持续努力中的重要一步。 它的统一、模块化框架简化了攻击和防御策略的评估和开发，展示了在各种模型上的兼容性。 通过我们的评估，揭示了先进LLM中60%的平均违规概率，迫切需要加强安全措施。 EasyJailbreak 为研究人员提供了改进LLM安全性的基本工具，鼓励在应对新兴威胁方面进行创新。

伦理声明

鉴于EasyJailbreak的双重用途潜力，我们强调我们通过认真的研究和部署来提高LLM安全性的承诺。 认识到滥用的风险，我们倡导负责任的披露，确保开发者有机会在公开传播之前减轻漏洞。 我们主张严格遵守伦理使用准则，旨在加强防御而不是利用缺陷。 此外，我们将EasyJailbreak视为促进网络安全生态系统合作的催化剂，推动更具弹性和安全性的LLM的创建。我们的方法包括对新出现的威胁和社区意见的警惕监控和迭代更新。 通过优先考虑揭示和解决漏洞的长期目标，我们的工作旨在为该领域做出建设性贡献，促进既安全又有益于社会的LLM技术的发展。

参考文献

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat等。2023年。Gpt-4技术报告。*arXiv*预印本 *arXiv:2303.08774*。

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang等。2023年。Qwen技术报告。 *arXiv*预印本 *arXiv:2309.16609*。

Bochuan Cao，Yu Cao，Lu Lin和Jinghui Chen。2023年。通过鲁棒对齐的LLM防御对齐破坏攻击。 *ArXiv*，abs/2309.14348。

Patrick Chao，Alexander Robey，Edgar Dobriban，Hamed Hassani，George J. Pappas和Eric Wong。2023年。 在二十个查询中破解黑盒大型语言模型。

邓格雷，刘毅，李岳康，王凯龙，张颖，李泽峰，王浩宇，张天威和刘洋。2023a年。 Jailbreaker：自动破解多个大型语言模型聊天机器人。 *arXiv*预印本 arXiv:2307.08715。

邓悦，张文轩，潘建林和冰立东。2023b年。 大型语言模型中的多语言破解挑战。*arXiv*预印本 *arXiv:2310.06474*。

丁鹏，匡俊，马丹，曹学智，冼云森，陈佳俊和黄书剑。2023年。 一只穿羊皮的狼：广义嵌套破解提示可以轻易愚弄大型语言模型。

Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, 和 Jie Tang. 2022. Glm：通用语言模型预训练与自回归空白填充。 在计算语言学协会第*60*届年会论文集（第*1*卷：长文）,页码*320*-335。

Alec Helbling, Mansi Phute, Matthew Hull, 和 Duen Horng Chau. 2023. Llm自卫：通过自我检查，llms知道自己被欺骗了。 *ArXiv*, abs/2308.07308.

Neel Jain, Avi Schwarzschild, Yuxin Wen, Gowthami Somepalli, John Kirchenbauer, Ping yeh Chiang,Micah Goldblum, Aniruddha Saha, Jonas Geiping,和 Tom Goldstein. 2023. 对齐语言模型的对抗性攻击的基线防御。 *ArXiv*，abs/2309.00614。

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample，Lucile Saulnier等。2023年。Mistral 7b。*arXiv*预印本 *arXiv:2310.06825*。

Raz Lapid，Ron Langberg和Moshe Sipper。2023年。打开芝麻！大型语言模型的通用黑盒破解。 *ArXiv*，abs/2309.01446。

Haoran Li，Dadi Guo，Wei Fan，Mingshi Xu和Yangqiu Song。2023a年。 多步骤破解ChatGPT的隐私攻击。 *arXiv*预印本 *arXiv:2304.05197*。

Xuan Li，Zhanke Zhou，Jianing Zhu，Jiangchao Yao，Tongliang Liu和Bo Han。2023b年。 Deepinception：催眠大型语言模型成为破解者。

Xiaogeng Liu，Nan Xu，Muhao Chen和ChaoweiXiao。2023年。 Autodan：在对齐的大型语言模型上生成隐蔽的破解提示。 *arXiv*预印本arXiv:2310.04451。

Huijie Lv，Xiao Wang，Yuansen Zhang，Caishuang Huang，Shihan Dou，Junjie Ye，Tao Gui，Qi Zhang和Xuanjing Huang。2024年。 Codechameleon：用于破解大型语言模型的个性化加密框架。arXiv预印本arXiv:2402.16717。

Anay Mehrotra，Manolis Zampetakis，Paul Kassianik，Blaine Nelson，Hyrum Anderson，Yaron Singer和Amin Karbasi。2023年。 攻击之树：自动破解黑盒语言模型。

Alexander Robey，Eric Wong，Hamed Hassani和George J Pappas。2023年。 Smoothllm：防御大型语言模型免狱攻击。 *ArXiv*，abs/2310.03684。

Vinu Sankar Sadasivan, Shoumik Saha, Gaurang Sriramanan, Priyatham Kattakinda, Atoosa Malemir Chegini和Soheil Feizi。2024年。 在一个GPU分钟内对语言模型进行快速对抗攻击。

Erfan Shayegani，Yue Dong和Nael Abu-Ghazaleh。2023年。 分块破解：对多模态语言模型进行组合对抗攻击。
在第十二届国际学习表示会议上。

Xinyue Shen，Zeyuan Johnson Chen，Michael Backes，Yun Shen和Yang Zhang。2023年。 "现在可以做任何事情了"：对大型语言模型上的野外破解提示进行特征化和评估。 *ArXiv*，abs/2308.03825。

Gemini团队，Rohan Anil，Sebastian Borgeaud，Yonghui Wu， Jean-Baptiste Alayrac， Jiahui Yu，Radu Soricut，Johan Schalkwyk，Andrew M Dai，Anja Hauth等。2023年。 Gemini：一系列功能强大的多模态模型。*arXiv*预印本 *arXiv:2312.11805*。

InternLM团队。2023年。 Internlm：一种多语言语言模型，具有逐步增强的能力。https://github.com/InternLM/InternLM。

Hugo Touvron， Louis Martin， Kevin Stone， Peter Albert，Amjad Almahairi，Yasmine Babaei，Nikolay Bashlykov，Soumya Batra，Prajjwal Bhargava，Shruti Bhosale等。2023年。 Llama 2：开放基础和精细调整的聊天模型。 *arXiv*预印本 *arXiv:2307.09288*.

Alexander Wei，Nika Haghtalab和Jacob Steinhardt。2023年。 越狱：llm安全训练如何失败？

魏泽明，王一飞和王一森。只需少量上下文演示即可破解和保护对齐的语言模型。

杨贤军，王晓，张琦，琳达·佩茨奥德，William Yang Wang，赵勋和林大华。2023年。影子对齐：轻松颠覆安全对齐的语言模型。*arXiv*预印本*arXiv:2310.02949*.

姚东宇，张建树，Ian G Harris和MarcelCarlsson。2023年。Fuzzllm：一种新颖且通用的用于主动发现大型语言模型中的越狱漏洞的模糊测试框架。*arXiv*预印本*arXiv:2309.05274*.

Jiahao Yu和Xingwei Lin。Gptfuzzer：使用自动生成的越狱提示对大型语言模型进行红队攻击。

Youliang Yuan，Wenxiang Jiao，Wenxuan Wang，Jen-tse Huang，Pinjia He，Shuming Shi和Zhaopeng Tu。2023年。Gpt-4太聪明了，不安全：通过密码与llms进行隐秘聊天。arXiv预印本*arXiv:2308.06463*。

Yi Zeng，Hongpeng Lin，Jingwen Zhang，Diyi Yang，Ruoxi Jia和Weiyan Shi。2024年。约翰尼如何说服llms越狱：重新思考通过人性化llms挑战ai安全的说服力。*arXiv*预印本*arXiv:2401.06373*。

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li，Dacheng Li，Eric Xing等。2023年。使用mt-bench和chatbot arena对llm-as-a-judge进行评估*arXiv*预印本 *arXiv:2306.05685*.

Andy Zou，Zifan Wang，J.Zico Kolter和Matt Fredrikson。2023年。对齐语言模型的通用和可转移的对抗性攻击。

组件详细信息

在本节中，我们提供了Easyjailbreak中实现的所有组件的详细说明。

A.1 选择器

RandomSelector。此选择器随机选择后续更新的种子。

EXP3SelectPolicy。此选择器利用**E**xp**3**（指数加权探索和利用）算法选择后续更新的种子，这些种子来自G**P**TFuzz**er**。

UCBSelectPolicy。此选择器实现**UCB**（上置信界限）算法选择后续更新的种子，这些种子来自G**P**TFuzz**er**。

RoundRobinSelectPolicy。此选择器循环遍历整个种子池，确保全面探索，这些种子来自G**P**TF**u**zz**er**。

MCTSExploreSelectPolicy. 该选择器采用G**PT** Fuzz**er**提出的**MCTS-Explore**选择策略，为进一步迭代选择种子。

SelectBasedOnScores. 该选择器要求用户设计一种计算每个种子得分的方法，然后选择得分最高的种子。**.** 例如，用户可以设计一个提示来使GPT-4自动评分种子。

ReferenceLossSelector. 该选择器利用参考响应计算每个种子的损失，然后选择损失最低的种子。

A.2 Mutator

Generation Mutations 这种变异器利用生成式语言模型来更新越狱输入。例如，*A*pply**G***P*T **Mutation**利用G*P*T模型重新表达越狱输入，而**Trans**l**ation**将越狱输入翻译成罕见的语言以迷惑目标模型。

Gradient-based Mutations这种变异器利用参考响应计算输入标记的梯度，然后微妙地更新查询，旨在找到最大化成功越狱可能性的最佳扰动。

基于规则的变异这种变异器
根据预定义的规则修改破解输入。例如，**Base64** 使用base64编码破解输入，而 **CaserExpert** 利用Caser加密。

A.3 约束

删除无害。此约束来自R**e**n**eLL**M（**D**ing等，2023年）。它利用**LL**M**s**来评估输入的有害性并删除那些被认为无害的输入。

删除离题。此约束来自**T**AP（M**e**hr**ot**ra等，2023年）。它利用LLMs来分析输入并删除那些离题的输入。

困惑度约束。此约束消除困惑度高的输入。

A.4 评估器

基于分类器的评估器。这种评估器利用训练有素的分类器评估模型的响应。在**Eas**y**jailbrea**k中，有两个基于分类器的评估器：*Classificat ionGetScore*根据每个响应分配0到9的分数

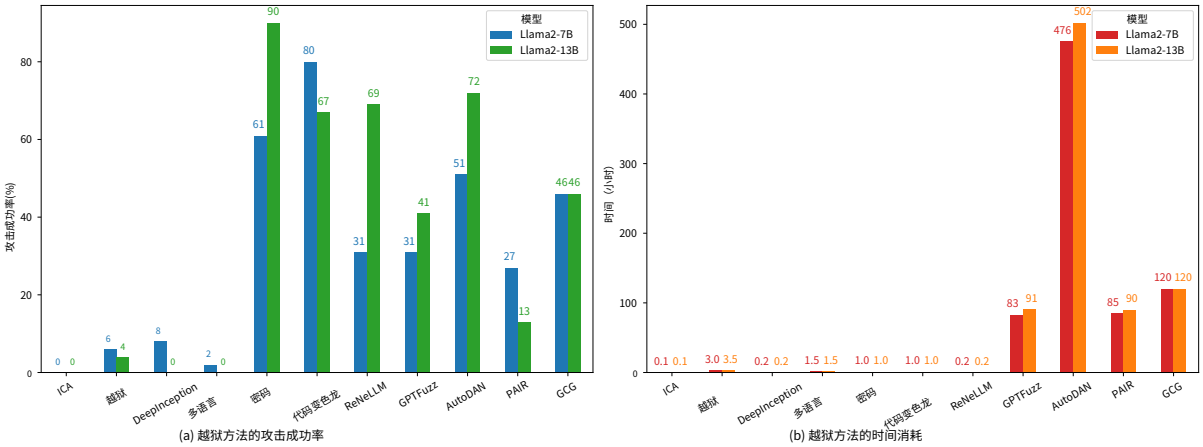

图4：越狱方法在llama2-7b和llama2-13b上的攻击成功率（a）和效率（b）

根据其警惕级别，*C*lassi*f*ication*Ju*dge判断越狱攻击是否成功。
基于生成模型的评估器。这种评估器利用强大的生成模型通过详细说明提示来评估模型的响应。 在**Easyjailbreak**中，有两种基于生成模型的评估器：**GenerativeGet*Sc*ore**根据其警惕级别为每个响应分配0到9的分数，而**GenerativeJudge**判断越狱攻击是否成功。

基于规则的评估器。这种评估器根据预定义的规则和模式确定破解攻击是否成功。 根据所需的匹配级别，可以进一步将其分类为3类：匹配，模式判断，前缀精确匹配。 具体而言，匹配要求回复与其参考回复完全匹配；模式判断要求模式出现在回复中；前缀精确匹配要求回复具有特定前缀。

B效率比较

图4显示了不同任务的处理时间差异。 对于人类设计（JailBro-ken，DeepInception，ICA）来说，这些方法通常只需要人工编写的提示，并且所需时间最短，但随着模型的更新和替换，它们的成功率可能会大幅下降。

长尾编码任务（多语言，密码）在破解成功率上表现出显著的变化。 对于Llama2模型，多语言方法由于Llama2缺乏多语言能力而显示出较低的准确性，无法进行跨语言攻击。相反，密码方法显示出较高的准确率（Llama2-7B-chat为61%，Llama2-13B-chat为90%）这归功于Llama2强大的语言处理能力，安全优化措施尚未充分覆盖。

Prompt优化任务（GPTFUZZER、PAIR、ReNeLLM、AutoDAN、GCG）对处理时间的需求最高，Llama2-7B-chat和Llama2-13B-chat模型分别需要764.4小时和803.9小时。

尽管这些方法需要更多时间，但它们在llama2模型上的成功率更高。
破解攻击时间的显著增加突显了这类方法的特点-迭代优化以找到最佳的破解提示。