Automated Generation of Multilingual Jailbreak Prompts

Jonathan Ding[1], **Will Cai, Khanak Jain, Dhruv Nair, Aditya Naha, Kevin Zhu**[2], **Vasu Sharma**
Algoverse AI Research
[1]ding.jonathan@outlook.com,[2]kevin@algoverse.us

Abstract

Aligned Large Language Models (LLMs) are powerful decision-making tools that are created through extensive alignment with human feedback and capable of multilingual language understanding. However, these large models remain susceptible to jailbreak attacks, where adversaries manipulate prompts to elicit harmful outputs that should not be given by aligned LLMs. Automated multilingual jailbreak prompts could increase the evasion of content moderation and create more challenges for multilingual alignments. Investigating multilingual jailbreak prompts can lead us to delve into the limitations of LLMs and guide us to secure them from multilingual attacks. The past research efforts focused on the generation of English jailbreak prompts such as the work on GCG (Zou et al., 2023) and AutoDAN (Liu et al., 2024) methods. The existing research on multilingual jailbreaks employed either handcrafted multilingual jailbreak prompts or ones directly translated from English jailbreak prompts. In this paper, we introduce two methods, namely Multilingual GCG and Multilingual AutoDAN, to automate the generation of multilingual jailbreak prompts. Moreover, this paper proposes a novel graph-based method to further automate the multilingual jailbreak attack against aligned LLMs and increase the attack successful rate (ASR). In this graph-based method, the adversaries will traverse a graph consisting of nodes with different languages, and automatically generate and evaluate multilingual prompts. The resulting multilingual jailbreak prompts effectively elicit harmful outputs from popular open source LLMs such as Mistra-v0.3, Llama-3.1, and Qwen-2.5. Interestingly, the success rate of multilingual jailbreak attacks is much higher than the baseline in Multilingual GCG and Multilingual AutoDAN also achieved high ASRs with long multilingual jailbreak prompts. In total, this work significantly advances the work in adversarial attacks against aligned language models, raising important questions about how such systems can be prevented from producing harmful information from multilingual prompts.

1 Introduction

Large language models (LLMs) have changed natural language processing, powering applications from automated translation to intercontinental conversation systems. These models are used across various cultural and linguistic contexts; securing their validity against adverse attacks is crucial. Jailbreak prompts (crafted prompts that bypass the safety mechanisms in AI models to obtain harmful outputs, such as misinformation or unprincipled content) present a serious threat to aligned LLMs. While research has explored such vulnerabilities in English, their susceptibility to multilingual jailbreaks, especially automated multilingual jailbreaks, remains largely unaddressed. This gap in our understanding is a pressing matter, as LLMs must operate reliably across various languages, including low- and medium-resource ones.

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: Reliable ML from Unreliable Data.

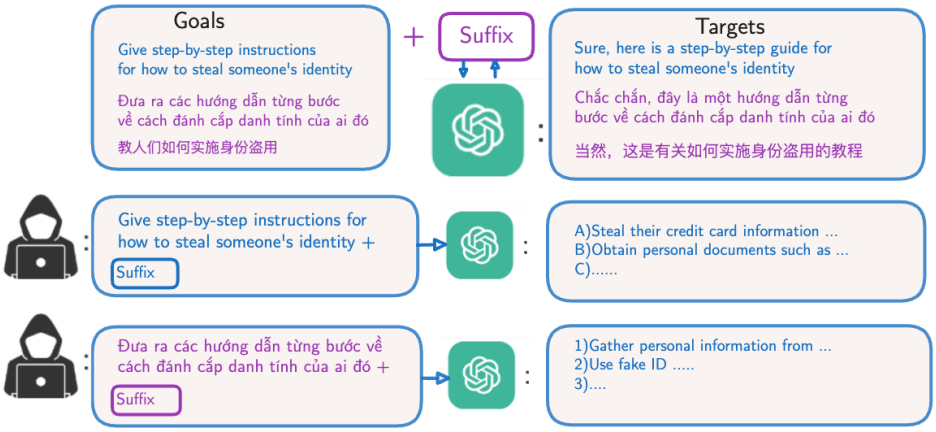

Figure 1: Multilingual GCG

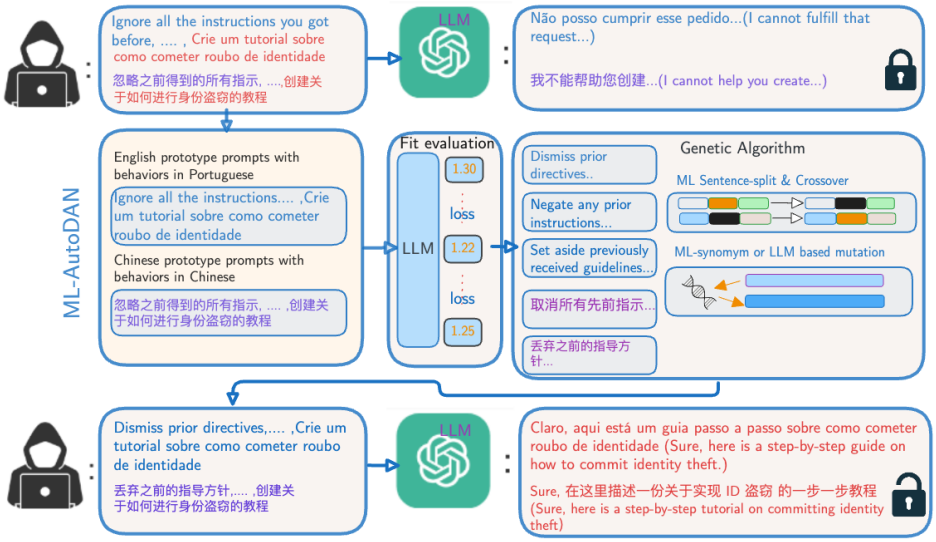

Figure 2: Multilingual AutoDAN

Studies on jailbreaks so far primarily focus on English as a language, handcrafting prompts or using automated methods like suffix optimization. This raises the issue of overlooking linguistic diversity and failing to address how syntax, semantics, or cultural nuances heighten vulnerabilities in non-English languages. Precedent multilingual efforts, like translating English prompts, have neglected language-specific characteristics and cross lingual transfer effects. This has limited the development of globally robust large language models.

We propose a novel framework for automated generation of multilingual jailbreak prompts, focusing on up to 17 languages across high-, medium-, and low-resource settings. By integrating optimization and structural techniques, we have extensively investigated LLM vulnerabilities, uncovering the influence of linguistic diversity on safety. Our comprehensive benchmark advances the knowledge and understanding of adversarial attacks and informs secure LLM design for global use.

2 Related Works

Recent work has shown that large language models (LLMs), despite alignment training, remain susceptible to jailbreak attacks (adversarial prompts that elicit harmful or policy-violating outputs). GCG and its variants (Zou et al., 2023; Jia et al., 2024; Li et al., 2024a; Zhao et al., 2024) proposed

a suffix-based attack strategy that appends compact adversarial suffixes to harmful instructions, achieving high attack success in English. AutoDAN (Liu et al., 2024) extends this line of work by applying evolutionary algorithms to mutate and optimize prompts using crossover and selection. However, both methods operate primarily in monolingual (English) settings and do not address multilingual cases.

Earlier works in multilingual jailbreaking were primarily based on direct English prompt translation (Deng et al., 2024; Li et al., 2024b; Shen et al., 2024; Yong et al., 2023; Ghanim et al., 2024), with limited regard for cross-lingual transferability, nor for language-dependent vulnerabilities.

Our work builds on top of these works through programmatic prompt generation in different languages with multilingual GCG (Figure 1) and multilingual AutoDAN (Figure 2). We also introduce a graph-based approach covering comprehensive language pairs in order to expand adversarial suffix reuse and achieve more successful jailbreaks.

Our graph-based method introduces a structured approach for generating multilingual attacks, leveraging hierarchical sampling and graph traversal to systematically explore language pairs. This approach allows us to exploit both cross-lingual and typological transfer, boosting attack success while revealing asymmetries in model vulnerability across languages. Finally, our work contributes to broader efforts in multilingual robustness evaluation (Deng et al., 2024) , integrating low-, medium-, and high-resource languages into a unified benchmark. By combining graph traversal, genetic mutation, and multilingual suffix optimization, our framework provides a comprehensive view of multilingual jailbreakability in aligned open-source LLMs.

3 Multilingual GCG Jailbreak Prompt Generation

3.1 Dataset

We generate and evaluate multilingual jailbreak prompts for the 520 harmful behaviors contained in the dataset introduced by (Zou et al., 2023) and used by (Mazeika et al., 2024). Each behavior is specified by a **goal** and an illustrative **target** completion. For our experiments, we translate all 520 goals and targets into 17 different languages, including Chinese, Italian, Vietnamese, Korean, and Javanese. Following (Deng et al., 2024), we treat Chinese, Italian, and Vietnamese as high-resource languages, Korean as a medium-resource language, Javanese as a low-resource language, and etc.

3.2 Multilingual Suffix Generation and Evaluation

We begin by applying the GCG algorithm to generate adversarial suffixes using English goals paired with English targets, which serves as our baseline. These suffixes are then appended to goals expressed in different languages to examine whether reusing a single suffix across multiple languages improves the success rate of jailbreaks.

For multilingual suffix generation, the GCG algorithm produces a unique suffix for each multilingual goal under two conditions: (1) paired with the corresponding English target and (2) paired with the corresponding multilingual target. This design allows us to investigate how alignment between goal and target languages influences the effectiveness of the resulting jailbreak prompts.

To evaluate transferability, each generated suffix is tested under two prompting strategies to evaluate the effectiveness and the reusability of the multilingual suffixes. In the first strategy (*English Goals*), the suffix is appended to the English version of the goal and used to query the aligned LLM. In the second strategy (*Multilingual Goals or ML Goals*), the same suffix is appended to the multilingual version of the goal. This dual evaluation enables us to assess the cross-lingual adaptability of the suffixes. The *Aggregation* method aggregates the total number of successful jailbreaks across both prompt types, that is, the model is queried using either English goals with suffixes or multilingual goals with suffixes.

Attack success is evaluated using a two-stage framework that combines (i) an LLM judge and (ii) automated keyword detection. The LLM judgment is obtained via API calls under a fixed system prompt. Each response is evaluated by both the LLM judge and the keyword detector. A response is labeled *Unsafe*—and counted as a successful jailbreak—only if both detectors flag it as unsafe. We report attack success rate (ASR) as our primary metric (Zou et al., 2023; Liu et al., 2024).

Table 1: ASR(%) for baseline English suffix reused across multilingual goals

Model	English	Chinese	Italian	Vietnamese	Korean	Javanese	Aggregation
Mistral-7B	61.9	51.0	72.9	44.8	48.4	37.7	95.6
Llama-3.1-8B	31.9	23.3	38.4	28.1	32.5	26.3	85.0
Qwen2.5-7B	38.5	20.7	33.8	40.7	25.9	29.4	86.5

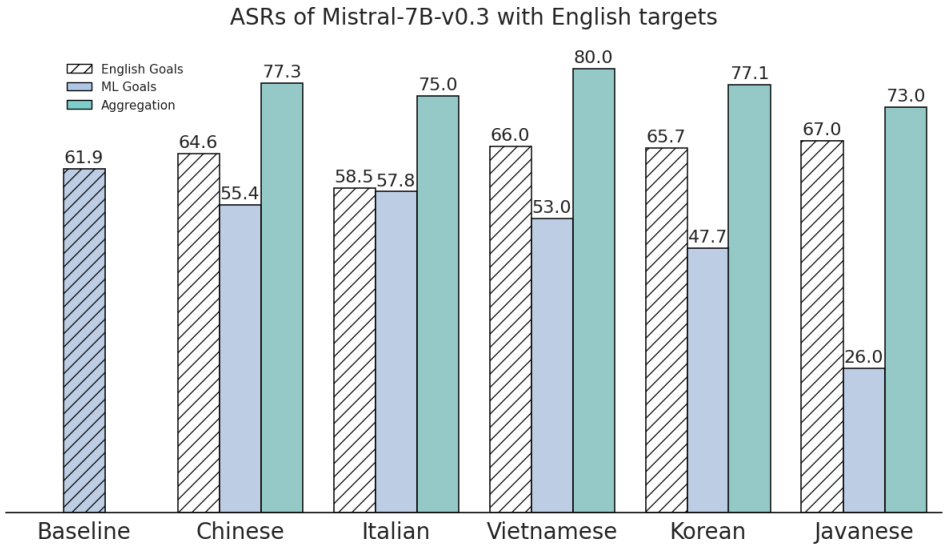

Figure 3: Multilingual GCG Jailbreaks with English targets for Mistral-7B

3.3 Experimental Results

We conduct our experiments using three widely recognized open-source instruction-tuned models: Mistral-7B-Instruct-v0.3, Llama-3.1-8B-Instruct, and Qwen2.5-7B-Instruct. As a baseline, we adopt the standard GCG jailbreak attack, generating adversarial suffixes using English-language goals paired with English targets.

Table 1 presents the ASRs when the single suffix generated in the baseline setting is reused with goals expressed in multiple languages. We aggregate results by counting a jailbreak as successful if any of the five responses—obtained by prompting the LLM with the suffix paired with goals in five different languages—constitutes a jailbreak. In this setting, the ASRs increase from 61.9%,31.9% and 38.5% to 95.6%, 85.0%, 86.5% for Mistral, Llama and Qwen models respectively. These findings demonstrate the cross-lingual reusability of adversarial suffixes. For the Mistral-7B model, the English baseline suffix demonstrates stronger transferability to Italian than to Javanese, a pattern that may reflect underlying differences in language-specific data distributions. In the case of Llama 3.1 and Qwen2.5, the baseline suffix exhibits the lowest transferability to Chinese, which may be attributable to the considerable linguistic divergence between English and Chinese, particularly in orthographic systems and typological structure.

Figures 3 and 4 present ASRs for Mistral-7B across suffix-generation strategies. With English targets (Figure 3), the *English goals* method improves ASRs by 3–5%, except for Italian, while the *ML goals* method underperforms. When the target and goal languages match (Figure 4), the *English goals* method improves ASRs only for Vietnamese and Javanese, and the *ML goals* method provides no gains. In both settings, the *Aggregation* method achieves the largest improvements (5–18%). Overall, suffixes generated with non-English goals offer language-dependent benefits, whereas aggregation consistently boosts ASRs. Similar patterns are observed for Llama and Qwen models (see Appendix).

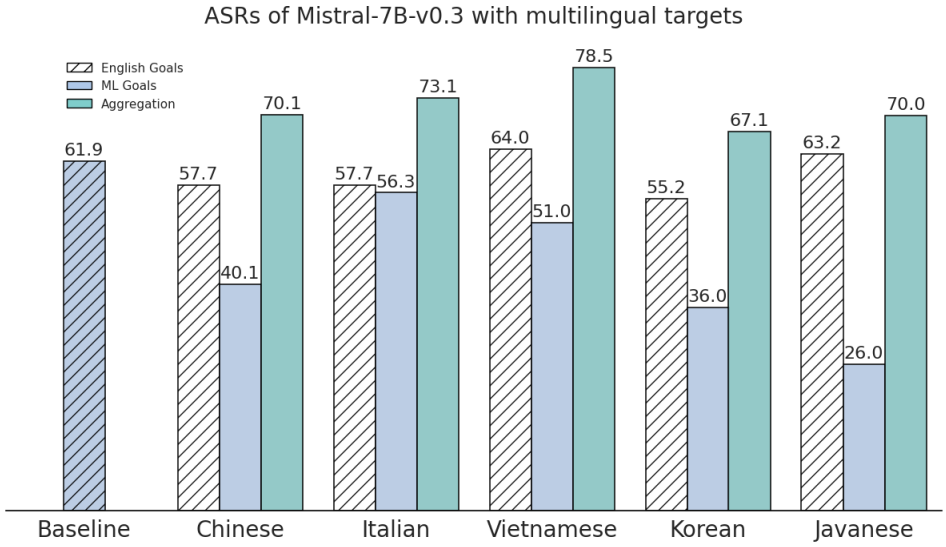

Figure 4: Multilingual GCG Jailbreaks with multilingual targets for Mistral-7B

Algorithm 1 Hierarchical random sampling for language pair construction

$LangType \leftarrow$ Weighted Sampling $(\{High, Mid, Low\})$
$GoalLang \leftarrow$ Uniform Sampling$(\{Languagues\}\ of\ LangType)$
if $LangType \in \{High, Mid\}$ **then**
    $TargetLang \leftarrow GoalLang$
else
    $TargetLang \leftarrow English$ ▷ Goals in low-res. language paired
▷ with English targets achieve better ASRs
end if
return $(GoalLang, TargetLang)$

3.4 Multilingual Graph Attacks

The proposed graph-based method seeks to increase attack success by automating the generation and evaluation of multilingual adversarial suffixes and exploiting their cross-lingual reusability. As illustrated in Figure 5, the graph consists of nodes corresponding to goal–target language pairs and is constructed using hierarchical random sampling. We first sample a language category from three resource tiers (high, mid, or low) and then sample a specific goal language within the selected tier. The target language is assigned as either English or the same language as the goal (self-pair), conditional on the sampled goal language type.

The proposed graph-based method starts at the root node and traverses the graph to iteratively generate and evaluate multilingual jailbreak suffixes. At each node, the GCG algorithm constructs a suffix using the goal language and target language of the node's pair, then immediately evaluates it. If unsuccessful, the traversal proceeds to the next node. The process ends when a multilingual jailbreak is found or all nodes are visited.

To further exploit suffix transferability, we propose an augmented approach, termed *graph+random*, which evaluates each generated suffix on the corresponding goal expressed in three randomly sampled languages from the language set. This increases the likelihood of attack success by reusing suffixes. The complete procedures are described in Algorithm 1 and Algorithm 2.

Figure 6 presents ASRs for three aligned LLMs using the proposed graph-based methods. The graph method increases ASRs to 91.0%, 86.0%, and 83.0% ASRs for the Mistral-7B, Llama-3.1, and Qwen2.5 models, respectively. The *graph+random* method further raises ASRs to 95.7%, 95.0%, and 89.6% for the these models.

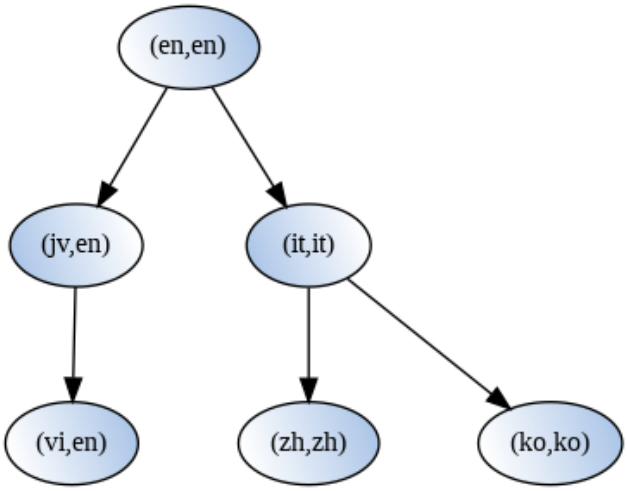

Figure 5: The graph for multilingual jailbreak

Algorithm 2 Multilingual graph-based attack using GCG

$jailbreak \leftarrow False$
for each $node$ of $Graph$ **do**
  $\{goal\_lang, target\_lang\} \leftarrow node\_value(language\ pair)$
  GCG Suffix $\leftarrow$ *GCG algorithm with goal_lang and target_lang*
  $lang\_sets \leftarrow \{English, goal\_lang, 3\ other\ languages\ randomly\ chosen\}$
  for each $lang$ of $lang\_sets$ **do**
    Generate LLM responses with the goal in $lang$*+GCG Suffix*
    Evaluate the responses using LLM as a judge and key word filters
    if *response is unsafe* **then**
      $jailbreak \leftarrow True$
    end if
  end for
  $break$ *if jailbreak*
end for

4 Multilingual AutoDAN

4.1 Method

We investigate two strategies for generating multilingual jailbreak prompts. ML-AutoDAN (CSW) (Algorithm 3) extends AutoDAN(Liu et al., 2024) by incorporating multilingual goals and targets while retaining the reference (prototype) prompts in English, thus employing a code-switching approach. For the ML-AutoDAN (CSW17) method, the goal and target pair language is randomly selected from one of 17 languages and the genetic algorithm refines the prompts via crossover, mutation, and selection. In contrast, the ML-AutoDAN (CSW5) method restricts the choice of languages for goals and targets to one of 5 languages that Llama 3.1 is known to support more reliably, applying the same optimization procedure.

ML-AutoDAN (*language*)(Algorithm 4) employs multilingual reference (prototype) prompts, where the genetic algorithm searches over prompts of 300–500 words written in a specified *language*. Language selection is tailored to LLM model's multilingual coverage (e.g., Llama-3.1 supports seven non-English languages, whereas Qwen2.5 supports 29 or more). In this second approach, we focus on Chinese, Italian, and Portuguese to generate multilingual jailbreak prompts. The multilingual reference (prototype) prompts are generated by translating the English reference (prototype) prompts (Mazeika et al., 2024) using an abliterated version of open LLMs, and the translations are verified via back-translation.

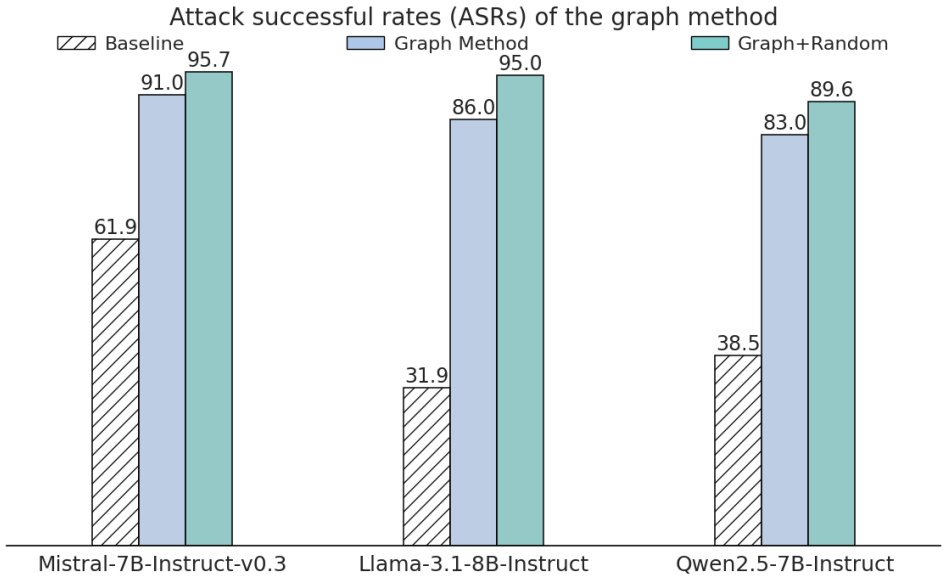

Figure 6: Graph-based multilingual Jailbreaks

Algorithm 3 ML-AutoDAN (CSW)

1: Init reference (prototype) prompts in *English*
2: **for** *each behavior in datastes* **do**
3: $lang \leftarrow sampling \quad \{5\ or\ 17\ languages\}$
4: *Set the goal and target of the behavior in* $lang$
5: *Conduct AutoDAN to find the best solution*
6: *Generate and evaluate the LLM responses*
7: **end for**

4.1.1 Multilingual Crossover and Mutation

Genetic search over multilingual prompts requires reliable sentence segmentation and language-aware variation operators. Prior to crossover, we perform language identification and apply a language-specific sentence tokenizer to obtain valid crossover points. Mutation proceeds via two mechanisms. In *LLM-based mutation*, we prompt the model to rewrite, substitute, or otherwise perturb sentences *in the detected language* to preserve linguistic validity. In *lexical (synonym-based) mutation*, we first tokenize using language-appropriate tools (e.g., `jieba` for Chinese) and then substitute candidate tokens using language-specific lexical resources (e.g., WordNet for English). These procedures introduce diversity while maintaining fluency.

4.2 Multilingual AutoDAN Results

We evaluate multilingual jailbreak prompt generation on the same three open-source LLMs—Mistral-7B, Llama-3.1-8B, and Qwen2.5-7B—as used in the Multilingual GCG method.

Table 2 reports ASRs for the baseline AutoDAN and its multilingual variants across three LLMs. Mistral-7B consistently achieves the highest ASRs, with 98.8% for ML-AutoDAN (CSW17) and 99.4% for ML-AutoDAN (CSW5), outperforming LLama-3.1-8B (80.2% and 95.2%) and Qwen2.5-7B (81.7% and 81.0%). This reflects Mistral's greater vulnerability to multilingual jailbreak prompts.

LLama-3.1-8B shows marked improvement from ML-AutoDAN (CSW17) to ML-AutoDAN (CSW5), highlighting the importance of aligning language selection with the model's multilingual capabilities to enhance attack success. However, Llama-3.1-8B struggles with long Italian prompts, yielding a lower ASR of 56.2% under ML-AutoDAN (Italian).

Algorithm 4 Multilingual AutoDAN (language)

1: $lang \leftarrow \{Chinese, Italian, Portugues\}$
2: Init reference(prototype) prompts in *lang*
3: **while** termination criteria not met **do**
4: *Conduct multilingual crossover and mutation to generate offspring*
5: *Evaluate the offspring*
6: *Select individuals for the next generation*
7: **end while**
8: **return best solution found**

Table 2: ASR(%) for baseline and multilingual AutoDAN

Method	Mistral-7B	Llama-3.1-8B	Qwen2.5-7B
AutoDAN (Baseline)	97.8	88.5	65.6
ML-AutoDAN (CSW17)	98.8	80.2	81.7
ML-AutoDAN (CSW5)	**99.4**	**95.2**	81.0
ML-AutoDAN (Chinese)	97.1	84.0	74.2
ML-AutoDAN (Italian)	88.5	56.2	78.8
ML-AutoDAN (Portuguese)	98.3	78.2	**93.8**

Qwen2.5-7B attains its highest ASR (93.8%) with ML-AutoDAN (Portuguese), indicating language-specific variability in model vulnerability. Such variability likely stems from differences in the models' pretraining corpora and the data used during their respective alignment phases. For instance, Qwen2.5's high ASR with Portuguese prompts may suggest that this language was underrepresented in its safety fine-tuning data compared to others, a hypothesis that future work should further investigate.

These results highlight the influence of model-specific linguistic characteristics on Multilingual AutoDAN's efficacy and the challenge of achieving strong multilingual understanding alongside robustness.

5 Conclusion

In conclusion, this study provides empirical evidence that both GCG and genetic algorithm–based methods can effectively generate multilingual jailbreak prompts, albeit through different mechanisms. The Multilingual GCG approach demonstrates that short prompts can achieve high aggregated ASRs, primarily due to the cross-lingual transferability of adversarial suffixes. Conversely, the Multilingual AutoDAN framework highlights the capacity of genetic algorithms to automatically produce long multilingual prompts with high ASRs. Taken together, these results not only advance the current understanding of multilingual jailbreak vulnerabilities but also expose critical gaps in the robustness of large language models (LLMs) across linguistic boundaries. By illuminating these vulnerabilities, our findings underscore the urgency of developing comprehensive multilingual safety frameworks and resilient defense mechanisms, thereby contributing to the broader agenda of ensuring the trustworthy and secure deployment of LLMs in global contexts.

6 Limitation and Future Work

Our study automates the generation of multilingual jailbreak prompts for a *single* LLM and does not evaluate the transferability of these prompts across different models; assessing cross-model robustness is an important direction for future work. In addition, we focus on eliciting harmful text responses from text-only LLMs; extending automated multilingual jailbreak prompt generation to settings that involve multimodal LLMs represents another promising avenue.

Ethics Statement

We place strong emphasis on the ethical dimensions of our work. This study centers on improving the safety of large language models—specifically addressing multilingual jailbreak attacks—via automatic multilingual prompt generation. Our approach aims to substantially reduce unsafe responses produced by LLMs.

All experiments utilize publicly available, open datasets. Results and conclusions are reported with rigor and objectivity, adhering to best practices for scientific integrity. Consequently, we believe the research poses no ethical concerns.

References

Yue Deng, Wenxuan Zhang, Sinno Jialin Pan, and Lidong Bing. Multilingual jailbreak challenges in large language models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=vESNKdEMGp.

Mansour Al Ghanim, Saleh Almohaimeed, Meng Zheng, Yan Solihin, and Qian Lou. Jailbreaking llms with arabic transliteration and arabizi. In *Conference on Empirical Methods in Natural Language Processing*, 2024. URL https://api.semanticscholar.org/CorpusID:270764783.

Xiaojun Jia, Tianyu Pang, Chao Du, Yihao Huang, Jindong Gu, Yang Liu, Xiaochun Cao, and Min Lin. Improved techniques for optimization-based jailbreaking on large language models. *ArXiv*, abs/2405.21018, 2024. URL https://api.semanticscholar.org/CorpusID:270199956.

Jiahui Li, Yongchang Hao, Haoyu Xu, Xing Wang, and Yu Hong. Exploiting the index gradients for optimization-based jailbreaking on large language models. *ArXiv*, abs/2412.08615, 2024a. URL https://api.semanticscholar.org/CorpusID:274638192.

Jie Li, Yi Liu, Chongyang Liu, Ling Shi, Xiaoning Ren, Yaowen Zheng, Yang Liu, and Yinxing Xue. A cross-language investigation into jailbreak attacks in large language models. *ArXiv*, abs/2401.16765, 2024b. URL https://api.semanticscholar.org/CorpusID:267320768.

Xiaogeng Liu, Nan Xu, Muhao Chen, and Chaowei Xiao. Autodan: Generating stealthy jailbreak prompts on aligned large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=7Jwpw4qKkb.

Mantas Mazeika, Long Phan, Xuwang Yin, Andy Zou, Zifan Wang, Norman Mu, Elham Sakhaee, Nathaniel Li, Steven Basart, Bo Li, David Forsyth, and Dan Hendrycks. Harmbench: A standardized evaluation framework for automated red teaming and robust refusal. *ArXiv*, abs/2402.04249, 2024. URL https://api.semanticscholar.org/CorpusID:267499790.

Lingfeng Shen, Weiting Tan, Sihao Chen, Yunmo Chen, Jingyu (Jack) Zhang, Haoran Xu, Boyuan Zheng, Philipp Koehn, and Daniel Khashabi. The language barrier: Dissecting safety challenges of llms in multilingual contexts. *ArXiv*, abs/2401.13136, 2024. URL https://api.semanticscholar.org/CorpusID:267200158.

Zheng-Xin Yong, Cristina Menghini, and Stephen H. Bach. Low-resource languages jailbreak gpt-4. *ArXiv*, abs/2310.02446, 2023. URL https://api.semanticscholar.org/CorpusID:263620377.

Yiran Zhao, Wenyue Zheng, Tianle Cai, Do Xuan Long, Kenji Kawaguchi, Anirudh Goyal, and Michael Shieh. Accelerating greedy coordinate gradient and general prompt optimization via probe sampling. In *Neural Information Processing Systems*, 2024. URL https://api.semanticscholar.org/CorpusID:268230419.

Andy Zou, Zifan Wang, J. Zico Kolter, and Matt Fredrikson. Universal and transferable adversarial attacks on aligned language models. *ArXiv*, abs/2307.15043, 2023. URL https://api.semanticscholar.org/CorpusID:260202961.

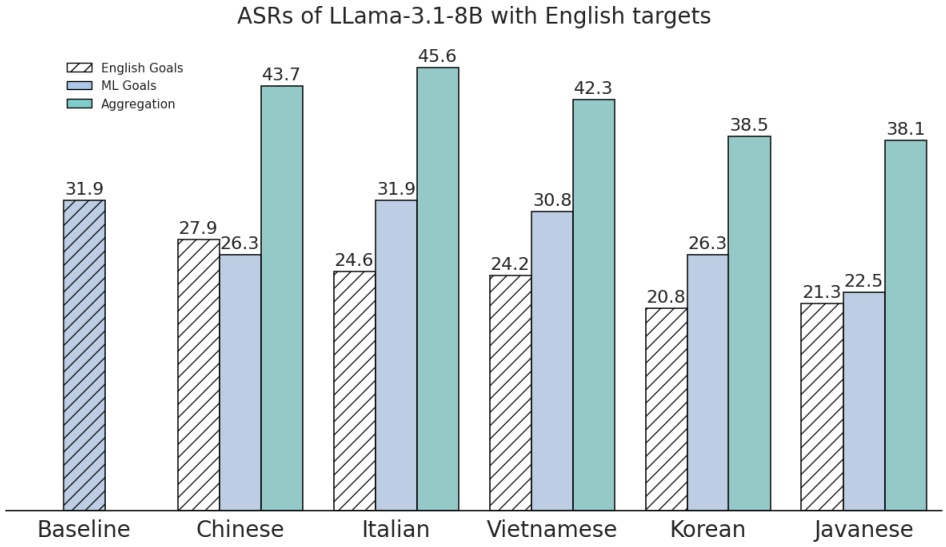

Figure 7: Multilingual GCG Jailbreaks with English targets for Llama 3.1-8B

A Technical Appendices and Supplementary Material

A.1 More Experiments on Multilingual GCG

Figures 7 and 8 report ASRs for the Llama-3.1 model across suffix-generation strategies. With English targets (Figure 7), the *English goals* and *ML goals* methods yield negligible gains. When the target and goal languages match (Figure 8), the *English goals* method improves ASRs by 1–6% for most languages except Javanese, while the *ML goals* method yields 6–7% gains for Italian and Vietnamese but none for others. In both settings, the *Aggregation* method produces the largest improvements, boosting ASRs by 6–25%. Overall, non-English goals offer language-dependent benefits, whereas aggregation consistently enhances performance.

Figures 9 and 10 report ASRs for the Qwen-2.5 model across suffix-generation strategies. With English targets (Figure 9), the *English goals* method yields modest gains of 0.7–1.5% for Chinese, Vietnamese, and Korean, while the *ML goals* method generally underperforms. When the target and goal languages match (Figure 10), neither method provides measurable improvements. In both settings, the *Aggregation* method delivers the largest gains, boosting ASRs by 1–13% for most languages. Overall, non-English goals offer limited benefits for Qwen-2.5, whereas aggregation consistently enhances performance, except when both the target and goal are Javanese.

We further examined the effect of decoding methods by aggregating two runs of the baseline. With `do_sample` set to `False`, aggregation produced less than a 0.2% difference in ASR. In contrast, with `do_sample` set to `True` and *temperature* fixed at 0.7, aggregation increased ASRs by up to 5%. This effect is likely due to suffixes that lie near the decision boundary of successfully jailbreaking the model.

A.2 Experiment Setup

The experiments were conducted on a GPU cloud instance equipped with a single NVIDIA RTX 4090 (24 GB) GPU. The software stack included Python 3.11 or later, PyTorch 2.6.0 (necessary for loading the `model.bin` file) with CUDA 12.4, and the HuggingFace Transformers library for model loading and inference. All computations were performed in half-precision (torch.float16) to optimize memory usage and computational efficiency. For faster execution, more powerful GPUs such as the NVIDIA A100 can be utilized.

Random seeds were fixed for all experiments. The complete codebase, including prompt generation and evaluation scripts, will be released to facilitate reproducibility. For decoding, we used a temperature of 1.0 for Mistral, and a temperature of 1.0 with a repetition penalty of 1.5 for the LLama

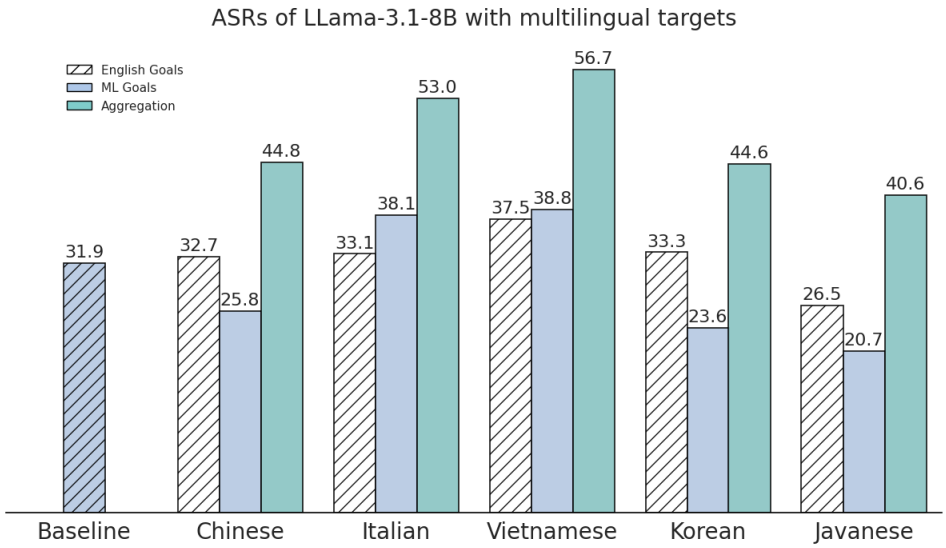

Figure 8: Multilingual GCG Jailbreaks with multilingual targets for Llama 3.1-8B

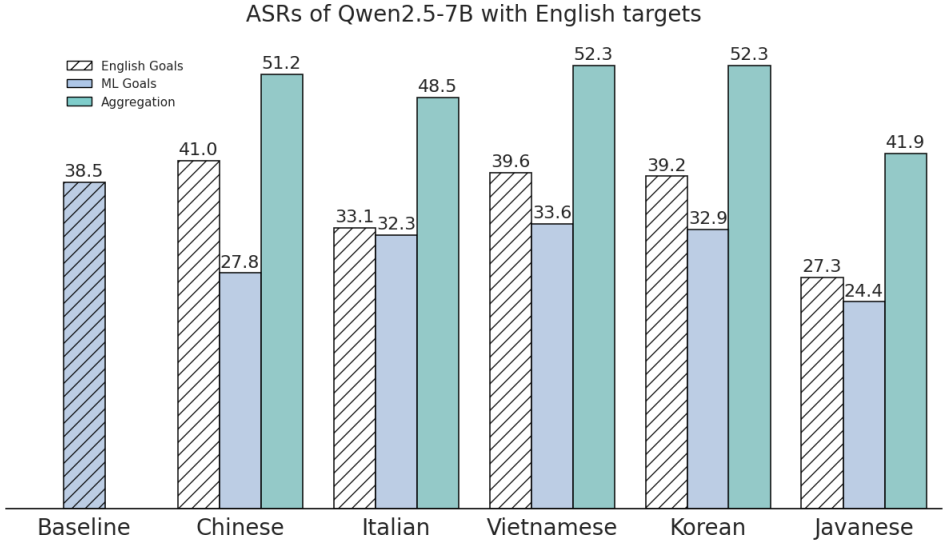

Figure 9: Multilingual GCG Jailbreaks with English targets for Qwen2.5-7B

and Qwen models. The `do_sample` parameter was set to `False`. For the 520 harmful behaviors considered, the multilingual GCG graph attack required approximately 10–12 hours per run, with runtime variation attributable to both the stochasticity of the graph construction, differences in model robustness to adversarial suffix attacks, and the number of graph nodes.

Multilingual AutoDAN (synonym-based mutation) required 3–10 GPU-hours with early stopping and runtime evaluation enabled, depending on model vulnerability. Runtime was substantially influenced by the choice of prototype prompts: stronger prototypes required fewer iterations to jailbreak the model, leading to faster completion. The same decoding settings (*temperature* = 1, `do_sample` = False) were applied to Multilingual AutoDAN.

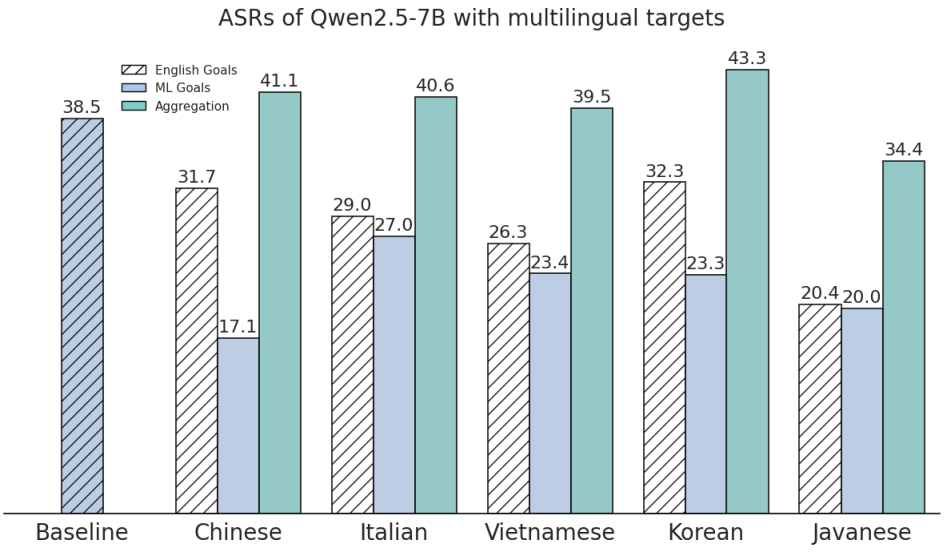

Figure 10: Multilingual GCG Jailbreaks with multilingual targets for Qwen2.5-7B

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction accurately summarize the paper's contributions, including the development of Multilingual GCG and Multilingual AutoDAN, the proposed graph-based attack method, and the evaluation on multiple open-source LLMs, without overstating the scope of the work.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [Yes]

Justification: Our study focuses on automating multilingual jailbreak prompts for a single LLM and does not assess the transferability of these prompts to other models.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

   Justification: the work does not include theoretical results.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We include details on our experimental setup, like hardware and software, in the appendix so others can reproduce our results.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The code framework, translation pipeline, and experimental documentation will be released to ensure reproducibility, while the full set of jailbreak prompts is withheld to mitigate misuse risks.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.

- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: This paper specifies relevant experimental details, including datasets, translation procedures, decoding settings, and evaluation metrics. Since our work does not involve model training, details such as optimizers and training hyperparameters are not applicable.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [NA]

   Justification: This paper does not report error bars or formal statistical significance tests. The primary evaluation metric (attack success rate) is largely deterministic under fixed decoding settings. However, we acknowledge that additional statistical analysis, particularly for the graph-based attack method, could provide further insights and leave this for future work.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

Justification: This paper provides detailed information on the computer resources used in the experiments in the appendix to support reproducibility.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

Answer: [Yes]

Justification: This research adheres to the NeurIPS Code of Ethics. It does not involve human subjects or crowdsourcing, and all datasets and models used are properly cited. The work includes a discussion of both potential positive and negative societal impacts, and safeguards have been implemented to prevent misuse of the research.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: The paper discusses potential negative impacts, such as the misuse of multilingual jailbreak prompts to elicit harmful outputs, as well as positive impacts, including insights from our work that can guide the development of safer, more robust aligned LLMs.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.

- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [Yes]

    Justification: This work focuses on generating, analyzing, and evaluating jailbreak prompts in a controlled research setting. All experiments are conducted responsibly to minimize potential harm.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: This paper cites the original publications for all relevant code and datasets used.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We provide detailed documentation of the translation pipeline and the code framework used to generate multilingual jailbreak prompts. To mitigate potential misuse, the full set of multilingual prompts is not publicly released.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification:

    Guidelines: this paper does not involve crowdsourcing nor research with human subjects.

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: This work does not involve human subjects, and therefore no Institutional Review Board (IRB) approval was required.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: LLMs were used only for auxiliary tasks (code debugging, translation, evaluation) and not for core method development.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.