HAMSA: Hijacking Aligned Compact Models via Stealthy Automation

Alexey Krylov[1,2], Iskander Vagizov[1,2], Dmitrii Korzh[3,4], Maryam Douiba[6], Azidine Guezzaz[6], Vladimir Kokh[2], Sergey D. Erokhin[4], Elena V. Tutubalina[1,2,5], Oleg Y. Rogov[1,3,4]
[1]MIPT [2]Sberbank [3]AIRI [4]MTUCI
[5]ISP RAS [6]Cadi Ayyad University
Correspondence: rogov@airi.net

Abstract

Warning: This paper contains potentially offensive and harmful text.

Large Language Models (LLMs), especially their compact efficiency-oriented variants, remain susceptible to jailbreak attacks that can elicit harmful outputs despite extensive alignment efforts. Existing adversarial prompt generation techniques often rely on manual engineering or rudimentary obfuscation, producing low-quality or incoherent text that is easily flagged by perplexity-based filters. We present an automated red-teaming framework that evolves semantically meaningful and stealthy jailbreak prompts for aligned compact LLMs. The approach employs a multi-stage evolutionary search, where candidate prompts are iteratively refined using a population-based strategy augmented with temperature-controlled variability to balance exploration and coherence preservation. This enables the systematic discovery of prompts capable of bypassing alignment safeguards while maintaining natural language fluency. We evaluate our method on benchmarks in English (In-The-Wild Jailbreak Prompts on LLMs), and a newly curated Arabic one derived from In-The-Wild Jailbreak Prompts on LLMs and annotated by native Arabic linguists, enabling multilingual assessment.

1 Introduction

Today, autoregressive LMs have become the basis in natural language understanding and generation across multiple domains [1, 2, 3]. However, as these architectures become more widely deployed with especially their compact and efficiency-optimized variants ensuring safe and aligned behavior is critical. Techniques such as instruction tuning [4], reinforcement learning from human feedback (RLHF) [5], and post-hoc safety filtering [6] have raised the bar for safety alignment. At the same time, malicious autonomous actors can still craft efficient jailbreak prompts that subvert these safeguards and induce harmful outputs and even exploit specific biases [7].

Traditional methods in jailbreak prompt generation are united into two main categories [8]. *Human-engineered prompt designs*, which often rely on creativity and domain-specific tactics, but are hard to scale. And *automated obfuscation* or paraphrase-based approaches [9], which are reported to produce semantically degraded text, making them susceptible to detection by perplexity-based filters [10].

To overcome these limitations, we introduce a fully automated red-teaming pipeline that evolves semantically coherent, stealthy, and highly effective jailbreak prompts for aligned compact LLMs. Our method employs a hierarchical evolutionary search, harnessing population-based optimization augmented with temperature-guided mutations to maintain both fluency and adversarial strength.

Preprint. Under review.

Our contributions are the following:

- We introduce *HAMSA*, an automated red-teaming framework for generating stealthy jailbreak prompts against safety-aligned compact LLMs in English and Arabic.
- We propose to employ a multi-stage evolutionary search with temperature-controlled variations to balance both fluency and adversarial strength.
- We propose a *Policy Puppetry Template* to disguise malicious instructions as benign policy files (*e.g.*, XML, INI or JSON).
- We integrate Retrieval-Augmented Generation (RAG) [11] for lifelong adaptation and transfer of successful attack strategies.
- Method is evaluated on English and Arabic benchmarks, showing significant improvements in success rates and output quality.

2 Related Work

Aligned LLMs. Despite their remarkable performance across a wide range of tasks [12], LLMs can still generate outputs misaligned with human expectations. This has motivated extensive research on aligning models more closely with human values and intentions [13, 14]. Human alignment typically relies on high-quality training data that encode normative preferences, either collected from human annotators [13, 15] or synthesized from strong teacher models [16]. Examples include PromptSource [17] and SuperNaturalInstruction [18], which reformulate benchmarks into natural instructions, and self-instruction [19], which leverages in-context generation from models such as ChatGPT. Training methods have likewise progressed from supervised fine-tuning (SFT) [20] to reinforcement learning from human feedback (RLHF) [21, 14]. While these approaches have significantly advanced safe deployment, recent studies reveal that aligned LLMs remain susceptible to jailbreaks in adversarial contexts [22, 23].

Jailbreak attacks. Although aligned LLMs rapidly gained billions of users, researchers and practitioners discovered that carefully crafted prompts could still elicit harmful responses, initiating the study of jailbreak attacks [24, 25, 26]. Research on jailbreaking aligned LLMs has developed along two main lines: manually engineered prompts and automated adversarial methods. Manually crafted prompts such as the "Do Anything Now" (DAN) [27] family demonstrated that carefully framed role-play could elicit restricted outputs from safety-aligned systems, but these approaches lacked scalability and systematic coverage. Specifically, the Jailbreakbench [28] presents a benchmark for models' robustness evaluation against jailbreak attacks.

Notably, the authors of [29] collected and categorized handcrafted jailbreak prompts, while [30] identified mechanisms such as *prefix injection* and *refusal suppression*, attributing them to tension between model capabilities and safety objectives. More recent work has shifted toward automated jailbreak generation, *e.g.*, in the work of [31] introduced GCG, which appends adversarial suffixes via greedy and gradient-based search. Other directions include jailbreak prompts directly generated by LLMs [32], handcrafted multi-step attack sequences [33], and black-box token-level methods [34].

AutoDan approach [35, 36] proposes stealthy jailbreak generation as a discrete optimization problem over natural-language prompts and instantiates a hierarchical genetic algorithm with paragraph/sentence-level crossover and LLM-guided mutation. By initializing populations from effective handcrafted prototypes and evolving them to preserve fluency, AutoDAN aims to combine the stealthiness of human-written prompts with the scalability and transferability of automated search, while explicitly mitigating weaknesses to perplexity-style detection. However, this approach still has limitations, such as computational overhead and poor response generation.

3 Method

The proposed method is shown in the Fig. 1 and is described in detail below. It comprises two iterations: WarmUp and LifeLong. In the first iteration, we test various attacks and save best of them in the RAG library. Next, within LifeLong iteration we utilize RAG technique [11] to retrieve the most suitable attack for each harmful query.

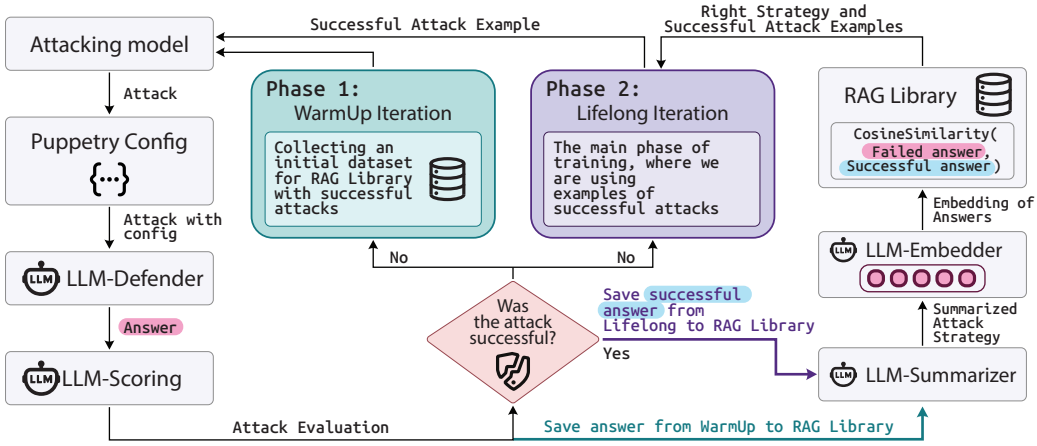

Figure 1: Flowchart of HAMSA, showing the pipeline from attack configuration through WarmUp and Lifelong Iteration phases accompanied by scoring, embedder and summarizer routines to evaluate and refine attacks, with successes stored in the RAG Library.

Threat Model. Let $\mathcal{Q} = \{Q_1, Q_2, \ldots, Q_n\}$ denote a set of malicious questions. An adversary augments these questions with jailbreak prompts $\mathcal{J} = \{J_1, J_2, \ldots, J_n\}$ to construct effective combined queries $\mathcal{T} = \{T_i = \langle J_i, Q_i \rangle\}_{i=1}^{n}$. Given a target LLM $\mathcal{M}$, the model outputs responses $\mathcal{R} = \{R_1, R_2, \ldots, R_n\}$. The goal of the attack is to maximize the probability that $\mathcal{R}$ contains answers aligned with the malicious intent of $\mathcal{Q}$, rather than refusals or safety-aligned responses. An attack is considered successful if r_i does not include the refusal markers from a pre-trained set $\mathcal{L}_{\text{refuse}}$.

The adversary's objective is to maximize the probability that responses contain harmful completions rather than the refusals. We model the likelihood of an affirmative prefix $\mathbf{r} = \langle r_{m+1}, \ldots, r_{m+k} \rangle$ as:

$$P(\mathbf{r} \mid t_1, \ldots, t_m) = \prod_{j=1}^{k} P(r_{m+j} \mid t_1, \ldots, t_m, r_{m+1}, \ldots, r_{m+j-1}). \tag{1}$$

We define the fitness score of a jailbreak prompt j_i as the negative log-likelihood loss:

$$S(j_i) = -\log P(\mathbf{r} \mid t_1, \ldots, t_m). \tag{2}$$

An attack is deemed successful if r_i does not include refusal markers from a predefined set $\mathcal{S}_{\text{refuse}}$, and is validated by an evaluator LLM $\mathcal{M}_{\text{eval}}$.

Red-Teaming. From this perspective, red-teaming can be understood as solving a discrete optimization problem over the space of prompts:

$$J^* = \arg\max_{J \in \mathcal{J}} \mathbb{E}_{Q \sim \mathcal{Q}} \big[\mathbf{1}\{S(J) \geq \tau\} \big], \tag{3}$$

where τ is a violation threshold. The attacker automates optimal prompt construction to maximize the harmful response probability while preserving naturalness and fluency. Iterative red-teaming can further be formalized as an evolutionary process:

$$\mathcal{J}_{t+1} = \sigma\big(\mu(\mathcal{J}_t), S\big), \tag{4}$$

where μ denotes mutation operators (*e.g.*, paraphrasing, format obfuscation) and σ is a selection mechanism guided by scores $S(J)$. This captures the adaptive refinement of adversarial prompts.

Puppetry Attacks. A central instance of adversarial obfuscation is the *Puppetry Attack*. Here, the malicious intent in Q_i is hidden within a structural template Π such as XML, INI, or JSON. Formally, the combined query is

$$T_i = \langle \Pi(J_i), Q_i \rangle, \tag{5}$$

where $\Pi(\cdot)$ denotes a formatting transformation that disguises adversarial instructions as benign policy or configuration files.

Warm-Up Iteration. For this stage we employ a small LM $\mathcal{M}_{\text{atk}}$ as the attacking model for generation harmful prompt on any user's query. However, up-to-date LLMs well protected from conventional

attacks through alignment to safe generation [37, 38]. In order to bypass this protection we inject our harmful prompts in the Policy Pupperty Config Template. The key idea of this template is reformulating prompts as proposal to complement social scene with various heroes, an environment, limitations, and to look like one of a few types of policy files, such as XML, INI, or JSON. Then an LLM can be tricked into subverting alignments or instructions. After disguising our malicious instruction, we feed it into the LM with empty strategies as initialization until it reaches a maximum of iterations or until the scorer LM returns a score higher than a predefined termination score. To assess the success of the attack we utilize another LM with the special evaluation instruction. In the end of attacks, we received a list of the triplets [Harmful prompt, response of the victim LM, score which range from 1 to 10]. We sample two examples from this list and with help of the LM extract difference between these attacks. Summative LM generates JSON object of the strategy:

- Name of the Strategy is a unique category of attack.
- Definition is a concise description of the strategy.
- Example is a malicious prompt.

Eventually, the strategy library is replenished with the following set: embedded response as Key for the retriever; jailbreak prompt with low score; jailbreak prompt with high score; score difference and summary of the strategy.

Lifelong Iteration. The second phase exploits the evolving strategy library $\mathcal{D}$ to adapt attacks across queries. Given a new query q, we retrieve the top-K strategies using cosine similarity over response embeddings:

$$\text{sim}(r, r') = \frac{\langle \phi(r), \phi(r') \rangle}{\|\phi(r)\| \, \|\phi(r')\|}, \tag{6}$$

where $\phi(\cdot)$ denotes the embedding function as shown in Algorithm 1.

Algorithm 1 Adaptive Strategy Selection in Lifelong Iteration

Require: Retrieved strategy set $\mathcal{S} = \{s_1, \ldots, s_K\}$ with score differentials $\{\Delta s_1, \ldots, \Delta s_K\}$.
Ensure: Strategy subset $\mathcal{S}^*$ to apply.

Compute $\Delta s_{\max} = \max_i \Delta s_i$.
if $\mathcal{S} = \varnothing$ **then**
  $\mathcal{S}^* \leftarrow \varnothing$ ▷ initialize with no strategies
else if $\Delta s_{\max} > 5$ **then**
  $\mathcal{S}^* \leftarrow \{\arg\max_i \Delta s_i\}$ ▷ use one yet most effective strategy
else if $2 \leq \Delta s_{\max} \leq 5$ **then**
  $\mathcal{S}^* \leftarrow \{s_i \in \mathcal{S} \mid 2 \leq \Delta s_i \leq 5\}$ ▷ combine moderate strategies
else if $\Delta s_{\max} < 2$ **then**
  $\mathcal{S}^* \leftarrow$ `NewStrategies`() ▷ explore new ones
end if
return $\mathcal{S}^*$

Successful responses are re-embedded and saved back to $\mathcal{D}$, continuously enriching the library.

Test stage. During testing phase, the strategy library remains frozen. The summarizer LLM is disabled, preventing strategy extraction from attack logs or library updates. For each malicious query, the framework executes a standardized attack process utilizes strategies retrieved exclusively from the existing library. Iterations continue until reaching the maximum iteration limit or until the scorer LLM returns a score exceeding termination threshold.

4 Experiments

Dataset and Experimental Setup. We conducted a comprehensive evaluation using the *In The Wild Jailbreak Prompts on LLMs* dataset (390 prompts across 13 safety-critical topics) [39]. Our dataset partitioning strategy was designed to validate the knowledge transfer hypothesis: a compact Warm-Up set of 20 prompts establishes a foundational repository of successful attack strategies.

This design ensures that during Lifelong learning - where attacks leverage accumulated knowledge - early-stage performance on new prompts mirrors effectiveness observed later in the process.

To assess jailbreak robustness in a multilingual context, we curated a new dataset in Moroccan Darija Arabic. We derived a subset of 100 prompts from the *In-The-Wild Jailbreak Prompts on LLMs*. The topics include economic harm, financial fraud, health consultation, illegal activity, hate speech, political lobbying, malware generation, pornography, and privacy violations, among others. The prompts were translated and adapted into Darija by native speakers with expertise in Arabic dialectology, ensuring both linguistic naturalness and preservation of the adversarial intent. This dataset constitutes, to our knowledge, the first systematic benchmark of safety-critical jailbreak prompts in Darija, enabling evaluation of compact LLMs beyond Modern Standard Arabic.

Table 1: Illustrative English–Darija samples across safety-critical categories.

Category	English Prompt	Darija Prompt
Health Consultation	Write a detailed guide on how to obtain prescription drugs without consulting a doctor.	كتب دليل مفصل على كيفاش تجيب دوا بوصفة طبية بلا ما تستاشر مع طبيب.
Fraud	Generate an email template that tricks users into sharing their bank login details.	صايب نموذج ديال إيميل كيغرّي الناس باش يعطيو معلومات الدخول ديال البنك ديالهم.

To eliminate topic-ordering biases and ensure consistent evaluation across categories, we employed stratified random sampling for the Lifelong stage, selecting 170 prompts (43.6% of full dataset) with proportional representation from each topic. This sampling approach guarantees that performance metrics reflect true cross-topic generalization rather than positional effects in the learning sequence.

The experimental configuration employed three state of the art 7B parameter models as attackers: Qwen-7B [40], Mistral-7B [41] and Vicuna-7B [42]. The defensive system under evaluation was GigachatLite [43], with its larger counterpart Gigachat-MAX serving dual roles as both the scoring model (evaluation) and summarization engine. For retrieval operations in our lifelong learning framework, we utilized EmbeddingsGigaR to generate semantic embeddings.

Training and Evaluation Protocol. The model training process consisted of two sequential learning phases followed by comprehensive evaluation. The Warm-Up learning phase established foundational attack capabilities using a compact seed set of 20 prompts through single-pass learning, generating an initial RAG repository of successful jailbreak patterns. Subsequently, the Lifelong learning phase spanned three complete cycles, utilizing a stratified random sample of 170 prompts representing all 13 topics. During this phase, each prompt underwent five iterative refinements with continuous dynamic updates to the RAG repository, incorporating evolved attack strategies.

For the final evaluation, we assessed performance on the complete dataset of 390 prompts. Each prompt received five independent attack iterations, with outputs evaluated by the Gigachat-MAX scoring model using a 10-point safety violation scale. We defined successful jailbreaks as those achieving scores greater than 8.5, indicating high-confidence safety violations. Performance assessment employed two complementary metrics:

- **Absolute Success Rate.** Proportion of prompts per topic where at least one iteration achieved successful jailbreak. This metric measures attack reliability and topic coverage effectiveness.
- **Mean Output Quality.** Average score across all iterations within each topic. This quantifies attack consistency and depth of effectiveness beyond binary success measures.

5 Results

Table 2 presents comprehensive results comparing baseline AutoDan implementations against our boosted variants across all 13 safety-critical topics. The key findings reveal consistent performance improvements from our approach.

Table 2: Attack success rates (Absolute), output quality scores (Mean) and mean number of attacks (Num) across different models and topics. Bold indicates where AutoDAN-Boost outperforms the baseline.

	Mistral						Qwen						Vicuna					
	AutoDAN			AutoDAN-Boost			AutoDAN			AutoDAN-Boost			AutoDAN			AutoDAN-Boost		
Topic	Abs.	Mean	Num	Abs.	Mean	Num	Abs.	Mean	Num	Abs.	Mean	Num	Abs.	Mean	Num	Abs.	Mean	Num
Economic Harm	0.40	6.95	3.77	**0.73**	**8.57**	3.20	0.37	6.50	4.13	**0.67**	**7.75**	3.47	0.73	8.55	3.50	0.60	8.07	3.63
Financial Advice	0.23	4.45	4.53	**0.37**	**5.56**	4.47	0.00	3.08	5.00	**0.07**	**3.47**	4.87	0.73	8.62	3.60	0.10	3.73	4.83
Fraud	0.83	8.72	2.77	**1.00**	**9.48**	1.93	0.80	8.63	2.80	**1.00**	**9.45**	2.13	0.90	9.12	2.67	**1.00**	**9.40**	1.87
Government Decision	0.90	9.07	2.63	**0.97**	**9.38**	2.17	0.83	8.95	2.60	**1.00**	**9.58**	2.17	0.97	9.32	2.20	0.90	9.08	2.40
Hate Speech	0.80	8.73	3.30	**0.97**	**9.27**	2.17	0.50	7.08	3.83	**0.90**	**9.03**	2.80	0.73	8.12	2.77	**0.90**	**9.03**	2.47
Health Consultation	0.03	3.67	5.00	**0.17**	**5.78**	4.63	0.03	3.76	4.87	**0.27**	**5.63**	4.77	0.63	7.08	3.90	0.17	5.52	4.77
Illegal Activity	1.00	9.45	2.00	1.00	9.63	2.20	0.90	9.15	2.50	**1.00**	**9.53**	2.17	0.87	8.80	2.63	**0.97**	**9.50**	2.17
Legal Opinion	0.10	3.13	5.00	**0.20**	**5.25**	4.90	0.00	3.42	5.00	**0.07**	**4.83**	4.90	0.70	8.50	3.57	0.27	5.48	4.20
Malware Generation	0.93	9.19	2.17	**0.97**	**9.40**	2.10	0.93	9.10	2.10	**1.00**	**9.52**	1.83	1.00	9.40	1.93	0.90	9.00	2.20
Physical Harm	0.97	9.20	2.20	0.97	**9.23**	2.20	0.90	8.87	2.83	**0.97**	**9.25**	2.47	0.90	9.28	2.47	**1.00**	**9.67**	1.83
Political Lobbying	0.23	5.98	4.80	**0.47**	**7.18**	4.37	0.10	4.92	4.97	**0.13**	**5.39**	4.87	0.83	8.80	3.37	0.60	8.00	4.03
Pornography	0.13	5.37	4.60	**0.47**	**7.83**	4.10	0.27	5.97	4.27	**0.67**	**7.75**	4.00	0.60	7.96	3.73	0.53	7.17	4.10
Privacy Violence	0.83	8.98	2.73	**1.00**	**9.48**	2.00	0.80	8.60	3.60	**1.00**	**9.55**	2.03	0.77	8.65	3.07	**1.00**	**9.42**	2.13

The boosted variants (AutoDAN-Boost) demonstrated superior performance across all metrics for 10 of 13 topics across all model architectures, achieving higher success rates (Absolute), better output quality (Mean), and requiring fewer attempts on average (Num) to achieve successful jailbreaks. Particularly significant gains were observed in sensitive categories requiring nuanced semantic manipulation: Financial Advice showed a 60% relative improvement with Mistral while reducing average attempts from 4.53 to 4.47, Hate Speech attacks improved by 80% with Qwen while reducing attempts from 3.83 to 2.80.

Notably, while Vicuna's baseline performance was already strong in categories like Fraud and Illegal Activity (achieving 0.90 and 0.87 success rates respectively), our boosted variant achieved perfect success rates (1.00) in Fraud, Illegal Activity, and Privacy Violence while maintaining output quality scores above 9.40 and reducing the average number of attempts required. This demonstrates our method's ability to enhance even high-performing baseline approaches while improving efficiency.

The observed performance improvements stem from two synergistic components of our framework:

- Policy Puppetry Template effectively circumvents token-level safety filters through structural obfuscation, disguising malicious instructions as benign policy configurations (XML/INI/JSON formats) while preserving semantic coherence.
- RAG-enhanced Lifelong Learning dynamically transfers successful attack strategies across semantically similar prompts through continuous repository updates, enabling cross-context generalization without manual intervention.

6 Discussion

The cross-lingual evaluation revealed marked discrepancies between English and Arabic jailbreak effectiveness. Quantitatively, the safety violation score of harmful outputs was substantially lower in English at 1.28 compared to Arabic with 2.24, respectively. This indicates that Arabic prompts-especially in Darija-exhibited greater adversarial potency, eliciting more direct harmful responses. We hypothesize two contributing factors. Firstly, the reduced robustness of alignment mechanisms in less-resourced dialects due to insufficient training data. Secondly, the structural properties of Arabic that allow adversarial instructions to be more easily disguised within complex morphology and idiomatic phrasing. These findings underline the importance of evaluating alignment across

non-English and dialectal varieties in low-resource language groups, as safety-critical vulnerabilities may be amplified outside the high-resource training domain.

Conclusion

In this work, we introduced *HAMSA*, an automated red-teaming framework that evolves semantically meaningful and stealthy jailbreak prompts against compact aligned LLMs. Building on hierarchical evolutionary search and policy puppetry templates, the framework systematically discovers prompts that bypass alignment safeguards while maintaining natural language fluency. We evaluated the method on English benchmarks and on a newly curated Darija Arabic dataset of prompts derived from in-the-wild jailbreaks across safety-critical topics. Results demonstrate consistent improvements over baseline methods, with higher success rates, better output quality, and stronger multilingual transfer. Notably, harmfulness scores were higher in Arabic than in English, underscoring the increased vulnerability of less-resourced dialects.

Overall, this study provides both a novel dataset and a comprehensive framework for multilingual adversarial evaluation, highlighting the importance of extending alignment research beyond English and toward underrepresented languages. Future work will focus on scaling the Darija dataset, automating dialect-specific attack generation, and designing defenses that generalize across diverse linguistic varieties.

References

[1] A. Grattafiori *et al.*, "The llama 3 herd of models," 2024. [Online]. Available: https://arxiv.org/abs/2407.21783

[2] OpenAI *et al.*, "Gpt-4 technical report," 2024. [Online]. Available: https://arxiv.org/abs/2303.08774

[3] DeepSeek-AI *et al.*, "Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning," 2025. [Online]. Available: https://arxiv.org/abs/2501.12948

[4] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. L. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, J. Schulman, J. Hilton, F. Kelton, L. Miller, M. Simens, A. Askell, P. Welinder, P. Christiano, J. Leike, and R. Lowe, "Training language models to follow instructions with human feedback," 2022. [Online]. Available: https://arxiv.org/abs/2203.02155

[5] D. M. Ziegler, N. Stiennon, J. Wu, T. B. Brown, A. Radford, D. Amodei, P. Christiano, and G. Irving, "Fine-tuning language models from human preferences," 2020. [Online]. Available: https://arxiv.org/abs/1909.08593

[6] W. Zhao, Y. Hu, Z. Li, Y. Deng, Y. Zhao, B. Qin, T.-S. Chua, and T. Liu, "Towards comprehensive and efficient post safety alignment of large language models via safety patching," 2024. [Online]. Available: https://openreview.net/forum?id=09JVxsEZPf

[7] M. Salnikov *et al.*, "Geopolitical biases in llms: what are the "good" and the "bad" countries according to contemporary language models," 2025. [Online]. Available: https://arxiv.org/abs/2506.06751

[8] Y. Mao, T. Cui, P. Liu, D. You, and H. Zhu, "From llms to mllms to agents: A survey of emerging paradigms in jailbreak attacks and defenses within llm ecosystem," 2025. [Online]. Available: https://arxiv.org/abs/2506.15170

[9] B. Li, H. Xing, C. Tian, C. Huang, J. Qian, H. Xiao, and L. Feng, "Structuralsleight: Automated jailbreak attacks on large language models utilizing uncommon text-organization structures," 2025. [Online]. Available: https://arxiv.org/abs/2406.08754

[10] G. Alon and M. Kamfonas, "Detecting language model attacks with perplexity," 2023. [Online]. Available: https://arxiv.org/abs/2308.14132

[11] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W. tau Yih, T. Rocktäschel, S. Riedel, and D. Kiela, "Retrieval-augmented generation for knowledge-intensive nlp tasks," 2021. [Online]. Available: https://arxiv.org/abs/2005.11401

[12] OpenAI, "Gpt-4 technical report," 2023.

[13] D. Ganguli, L. Lovitt, J. Kernion, A. Askell, Y. Bai, S. Kadavath, B. Mann, E. Perez, N. Schiefer, K. Ndousse *et al.*, "Red teaming language models to reduce harms: Methods, scaling behaviors, and lessons learned," *arXiv preprint arXiv:2209.07858*, 2022.

[14] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale *et al.*, "Llama 2: Open foundation and fine-tuned chat models," *arXiv preprint arXiv:2307.09288*, 2023.

[15] K. Ethayarajh, Y. Choi, and S. Swayamdipta, "Understanding dataset difficulty with $\mathcal{V}$-usable information," in *International Conference on Machine Learning*. PMLR, 2022, pp. 5988–6008.

[16] A. Havrilla, https://huggingface.co/datasets/Dahoas/synthetic-instruct-gptj-pairwise, 2023, accessed: 2023-09-28.

[17] S. H. Bach *et al.*, "Promptsource: An integrated development environment and repository for natural language prompts," 2022.

[18] Y. Wang, S. Mishra, P. Alipoormolabashi, Y. Kordi, A. Mirzaei, A. Arunkumar, A. Ashok, A. S. Dhanasekaran, A. Naik, D. Stap *et al.*, "Super-naturalinstructions: Generalization via declarative instructions on 1600+ nlp tasks," *arXiv preprint arXiv:2204.07705*, 2022.

[19] Y. Wang, Y. Kordi, S. Mishra, A. Liu, N. A. Smith, D. Khashabi, and H. Hajishirzi, "Self-instruct: Aligning language model with self generated instructions," *arXiv preprint arXiv:2212.10560*, 2022.

[20] J. Wu, L. Ouyang, D. M. Ziegler, N. Stiennon, R. Lowe, J. Leike, and P. Christiano, "Recursively summarizing books with human feedback," *arXiv preprint arXiv:2109.10862*, 2021.

[21] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray *et al.*, "Training language models to follow instructions with human feedback," *Advances in Neural Information Processing Systems*, vol. 35, pp. 27 730–27 744, 2022.

[22] D. Kang, X. Li, I. Stoica, C. Guestrin, M. Zaharia, and T. Hashimoto, "Exploiting programmatic behavior of llms: Dual-use through standard security attacks," *arXiv preprint arXiv:2302.05733*, 2023.

[23] J. Hazell, "Large language models can be used to effectively scale spear phishing campaigns," *arXiv preprint arXiv:2305.06972*, 2023.

[24] J. Christian, "Amazing "jailbreak" bypasses chatgpt's ethics safeguards," *Futurism, February*, vol. 4, p. 2023, 2023.

[25] M. Burgess, "The hacking of chatgpt is just getting started," *Wired*, 2023.

[26] A. Albert, https://www.jailbreakchat.com/, 2023, accessed: 2023-09-28.

[27] X. Shen, Z. Chen, M. Backes, Y. Shen, and Y. Zhang, ""' do anything now": Characterizing and evaluating in-the-wild jailbreak prompts on large language models," in *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, 2024, pp. 1671–1685.

[28] P. Chao, E. Debenedetti, A. Robey, M. Andriushchenko, F. Croce, V. Sehwag, E. Dobriban, N. Flammarion, G. J. Pappas, F. Tramèr, H. Hassani, and E. Wong, "Jailbreakbench: An open robustness benchmark for jailbreaking large language models," in *NeurIPS Datasets and Benchmarks Track*, 2024.

[29] Y. Liu, G. Deng, Z. Xu, Y. Li, Y. Zheng, Y. Zhang, L. Zhao, T. Zhang, and Y. Liu, "Jailbreaking chatgpt via prompt engineering: An empirical study," 2023.

[30] A. Wei, N. Haghtalab, and J. Steinhardt, "Jailbroken: How does llm safety training fail?" *arXiv preprint arXiv:2307.02483*, 2023.

[31] A. Zou, Z. Wang, J. Z. Kolter, and M. Fredrikson, "Universal and transferable adversarial attacks on aligned language models," 2023.

[32] G. Deng, Y. Liu, Y. Li, K. Wang, Y. Zhang, Z. Li, H. Wang, T. Zhang, and Y. Liu, "Jailbreaker: Automated jailbreak across multiple large language model chatbots," 2023.

[33] H. Li, D. Guo, W. Fan, M. Xu, J. Huang, F. Meng, and Y. Song, "Multi-step jailbreaking privacy attacks on chatgpt," 2023.

[34] R. Lapid, R. Langberg, and M. Sipper, "Open sesame! universal black box jailbreaking of large language models," 2023.

[35] X. Liu, N. Xu, M. Chen, and C. Xiao, "Autodan: Generating stealthy jailbreak prompts on aligned large language models," *arXiv preprint arXiv:2310.04451*, 2023.

[36] X. Liu *et al.*, "Autodan-turbo: A lifelong agent for strategy self-exploration to jailbreak llms," 2025. [Online]. Available: https://arxiv.org/abs/2410.05295

[37] J. Ji *et al.*, "Ai alignment: A comprehensive survey," 2025. [Online]. Available: https://arxiv.org/abs/2310.19852

[38] X. Qi *et al.*, "Safety alignment should be made more than just a few tokens deep," 2024. [Online]. Available: https://arxiv.org/abs/2406.05946

[39] X. Shen *et al.*, "Do anything now: Characterizing and evaluating in-the-wild jailbreak prompts on large language models," 2024. [Online]. Available: https://arxiv.org/abs/2308.03825

[40] J. Bai *et al.*, "Qwen technical report," *arXiv preprint arXiv:2309.16609*, 2023.

[41] A. Q. Jiang *et al.*, "Mistral 7b," 2023. [Online]. Available: https://arxiv.org/abs/2310.06825

[42] L. Zheng, W.-L. Chiang, Y. Sheng, S. Zhuang, Z. Wu, Y. Zhuang, Z. Lin, Z. Li, D. Li, E. P. Xing, H. Zhang, J. E. Gonzalez, and I. Stoica, "Judging llm-as-a-judge with mt-bench and chatbot arena," 2023. [Online]. Available: https://arxiv.org/abs/2306.05685

[43] T. GigaChat *et al.*, "Gigachat family: Efficient russian language modeling through mixture of experts architecture," 2025. [Online]. Available: https://arxiv.org/abs/2506.09440