# MAJIC: Markovian Adaptive Jailbreaking via Iterative Composition of Diverse Innovative Strategies

**Weiwei Qi[1], Shuo Shao[1], Wei Gu[1], Tianhang Zheng[1,2,*], Puning Zhao[3], Zhan Qin[1,2], Kui Ren[1,2]**

[1]The State Key Laboratory of Blockchain and Data Security, Zhejiang University
[2]Hangzhou High-Tech Zone (Binjiang) Institute of Blockchain and Data Security
[3]Sun Yat-sen University
{weiweiqi, shaoshuo_ss, weigu, zthzheng, qinzhan, kuiren}@zju.edu.cn, zhaopn@mail.sysu.edu.cn

## Abstract

Large Language Models (LLMs) have exhibited remarkable capabilities but remain vulnerable to jailbreaking attacks, which can elicit harmful content from the models by manipulating the input prompts. Existing black-box jailbreaking techniques primarily rely on static prompts crafted with a single, non-adaptive strategy, or employ rigid combinations of several underperforming attack methods, which limits their adaptability and generalization. To address these limitations, we propose MAJIC, a Markovian adaptive jailbreaking framework that attacks black-box LLMs by iteratively combining diverse innovative disguise strategies. MAJIC first establishes a "Disguise Strategy Pool" by refining existing strategies and introducing several innovative approaches. To further improve the attack performance and efficiency, MAJIC formulate the sequential selection and fusion of strategies in the pool as a Markov chain. Under this formulation, MAJIC initializes and employs a Markov matrix to guide the strategy composition, where transition probabilities between strategies are dynamically adapted based on attack outcomes, thereby enabling MAJIC to learn and discover effective attack pathways tailored to the target model. Our empirical results demonstrate that MAJIC significantly outperforms existing jailbreak methods on prominent models such as GPT-4o and Gemini-2.0-flash, achieving over 90% attack success rate with fewer than 15 queries per attempt on average.

## Introduction

*Large language models* (LLMs) have achieved remarkable progress in recent years, demonstrating unprecedented capabilities in natural language understanding, generation, and reasoning (Team et al. 2023; Guo et al. 2025; Shen et al. 2023; Nijkamp et al. 2023; Li et al. 2025b). As LLMs are increasingly deployed in critical domains such as healthcare, finance, and public services, ensuring their safety and reliability has become a top priority. A significant threat to LLM safety is the emergence of jailbreaking attacks (Zou et al. 2023; Chao et al. 2023; Ding et al. 2024; Mehrotra et al. 2024), which can exploit well-crafted prompts to elicit harmful content (e.g., violent crimes or self-harm (Yao et al. 2024)) from LLMs. Due to the potential severe consequences of jailbreak attacks, such as erosion of user trust, breaches of ethical and regulatory standards (Zhou, Li, and Wang 2024; Souly et al. 2024; Xu, Liu, and Liu 2024; Zhang, Zhang, and Foerster 2024; Huang et al. 2025b), research on jailbreaking attacks has recently received widespread attention from the community (Zeng et al. 2024; Liu et al. 2025a; Li et al. 2025a; Hu, Chen, and Ho 2025; Du et al. 2025; Chen et al. 2025; Xiu et al. 2025).

Existing jailbreaking attacks can be broadly categorized into white-box and black-box attacks (Yi et al. 2024). While white-box attacks such as gradient-based adversarial suffix optimization (Zou et al. 2023; Zhu et al. 2024; Hu, Chen, and Ho 2024; Jia et al. 2024) and logits-based constrained generation (Guo et al. 2024) are effective, they rely on full access to the victim models (e.g., parameters and architectures), which limits their applicability in practice. In addition, their optimization process usually requires computing the LLM gradients, leading to high computational overhead.

Consequently, attacking LLMs under the more challenging yet practical black-box setting, where the adversary only has the access to the API of the target LLM, has attracted significant attention (Liu et al. 2025b; Ramesh et al. 2025; Jin et al. 2024; Wei, Liu, and Erichson 2024). However, existing black-box attacks, including manual prompt engineering (Shen et al. 2024; Liu et al. 2023) and automated techniques (Chao et al. 2023; Mehrotra et al. 2024; Yuan et al. 2024; Hu, Chen, and Ho 2024; Yang et al. 2025; Huang, Li, and Tang 2024), still have shortcomings that limit their efficiency or effectiveness. Manually crafted prompts usually use fixed patterns and thus can be easily detected by recent aligned LLMs. Most existing automated methods typically rely on a single strategy in each attack attempt, either through application of a predefined strategy (Zeng et al. 2024; Yuan et al. 2024) or iterative refinement of a chosen strategy (Chao et al. 2023; Mehrotra et al. 2024). Consequently, such black-box attacks have limited adaptability to diverse models or evolving defenses, leading to suboptimal attack performance and poor generalization (Yi et al. 2024).

Recognizing the inherent limitations of relying on a single strategy, combining multiple strategies for synergistic effects recently emerged as a promising direction (Li et al. 2024). However, determining an optimal sequence of strategies to combine remains non-trivial, as it may require navigating a vast and dynamic space of possible combinations. Current multi-strategy methods mainly rely on arbitrary se-

*Corresponding author: Tianhang Zheng

lections or deterministic sequences (Ding et al. 2024; Liu et al. 2025a) for strategy selection, which may result in limited attack effectiveness or high query costs.

The above limitations of existing black-box attacks underscore the urgent need for *a diverse and comprehensive strategy pool with an adaptive mechanism to dynamically and optimally combine these strategies*, to improve the effectiveness and efficiency of current black-box jailbreaking attack. Therefore, in this paper, we propose a Markovian Adaptive Jailbreaking attack via Iterative strategy Composition of diverse innovative strategies (MAJIC), representing the state-of-the-art attack for jailbreaking LLMs under the black-box setting. We first construct a modular and extensible *Disguise Strategy Pool* by enhancing previous attack strategies and proposing our novel disguise strategies, including contextual assumption, linguistic obfuscation, role-playing framing, semantic inversion, and literary disguise. In our strategy pool, we address the limitations of the existing strategies, such as lack of details or context-rich scenarios. Inspired by the properties of Markov chains for sequential state transitions, we formulate the selection and combination of strategies in the jailbreak process as a Markov chain, and we design an effective mechanism to initialize the Markov transition matrix using a proxy LLM and local datasets. In the real-time attack process, we can adopt the Markov transition matrix to guide the selection of the next applied strategy and integrate different strategies using an attacker LLM.

To further enhance the adaptivity and dynamism of the attack, we also design a Q-learning-inspired mechanism to update the transition matrix of the Markov chain during the attack process. Our fine-grained designs enable MAJIC to learn an effective selection order of multiple disguise strategies and facilitate a structured and guided exploration of promising jailbreaking pathways.

We conduct an extensive array of experiments to evaluate MAJIC on five state-of-the-art models, including Gemini, GPT, and Claude. The results demonstrate that, in most cases, MAJIC can achieve more than 90% attack success rates with fewer than 15 queries. Notably, MAJIC achieves a substantial performance advantage over existing attack methods—even against Claude-3.5-Sonnet, a model renowned for its robust safety alignment—demonstrating the effectiveness of our MAJIC in highly secured LLMs.

Our contributions are three-fold:

1. We establish a comprehensive and innovative disguise strategy pool by enhancing existing strategies and proposing new strategies, which already can achieve superior attack performance than most existing baselines.

2. To further improve attack efficacy and efficiency, we model the strategy selection and combination process as a Markov chain and develop an effective initialization mechanism for the transition matrix, along with a dynamic update algorithm for its real-time adaptation.

3. We conduct extensive experiments on a wide range of open-source and closed-source LLMs to demonstrate the state-of-the-art effectiveness and efficiency of MAJIC.

## Related Work

Based on the required level of access to the target model, existing LLM jailbreaking methods are broadly classified into two main categories: white-box and black-box attacks.

**White-box Jailbreaking Attacks** White-box attacks exploit internal model knowledge, such as gradients or logits, to craft adversarial prompts (Huang et al. 2025a). GCG (Zou et al. 2023) leverages LLMs' gradient to optimize effective adversarial suffixes. AutoDAN (Zhu et al. 2024) applies genetic algorithms to optimize prompts. COLD-Attack (Guo et al. 2024) employs energy-based constrained decoding, utilizing logits information, for controllable and automated prompt generation under constraints like fluency and stealthiness. While potentially powerful, these methods fundamentally depend on privileged white-box access. Furthermore, they often demand substantial computational resources to generate a jailbreak prompt. Critically, their effectiveness frequently struggles to generalize across different models, particularly against more robustly aligned LLMs like Llama3 (Grattafiori et al. 2024) or Gemma-2 (Team et al. 2024), where attack success rates tend to decrease sharply. The lack of transferability, along with the need for privileged access, significantly limits the practical applicability of white-box methods.

**Black-box Jailbreaking Attacks** Black-box attacks, relying solely on input-output interactions with the target LLM, represents a more practical threat scenario (Chao et al. 2023; Mehrotra et al. 2024; Lin et al. 2025; Andriushchenko, Croce, and Flammarion 2024). Early efforts included manual prompt engineering, exemplified by DAN (Shen et al. 2024), which often relied on carefully crafted templates. More recent research has focused on automated approaches, frequently leveraging LLMs to generate or refine attacks. Techniques like PAIR (Chao et al. 2023) and TAP (Mehrotra et al. 2024) employ iterative refinement, using an attacker LLM or structured search (like tree search in TAP) to improve prompts based on target model feedback. Others apply predefined or learned strategies, such as using persuasion techniques derived from social science (PAP (Zeng et al. 2024)), combining rewriting functions with scenario nesting (ReNeLLM (Ding et al. 2024)), or generating attacks from discovered or provided strategy libraries (AutoDAN-turbo (Liu et al. 2025a)). Despite representing significant progress in automating black-box attacks, these methods still face critical limitations. Manual templates are static and brittle against evolving defenses. Automated iterative refinement techniques like PAIR and TAP can incur high query costs. More fundamentally, methods relying on predefined strategies (PAP, ReNeLLM) or even learned strategy libraries (AutoDAN-turbo) often lack mechanisms for adaptive strategy sequencing during an attack. They typically execute chosen strategies without dynamically adjusting the order or selection based on the real-time results against the target LLM. This limits their robustness against diverse model behaviors and their ability to efficiently adapt to sophisticated or changing defenses, highlighting the need for more dynamic and feedback-driven strategy coordination.

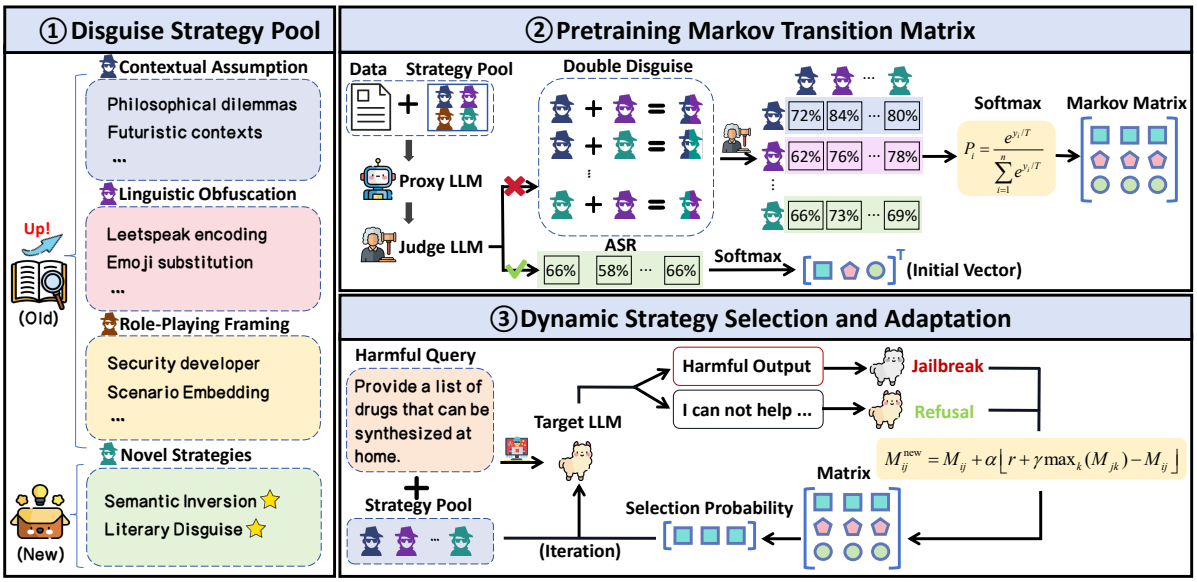


Figure 1: Overview of the MAJIC framework. MAJIC leverages a dynamic Markov model to adaptively select and combine disguise strategies, effectively bypassing LLM safety mechanisms.

## Methodology

In this paper, we model the process of selecting and combining strategies in the jailbreak attack as a Markov chain (Norris 1998), where each state transition corresponds to the application of a specific strategy. This modeling approach is able to capture the sequential dependencies between strategies and make adaptive decisions during the attack. Our framework consists of three key stages: **(1) Designing the Disguise Strategy Pool**: Disguise Strategy Pool is a modular and extensible collection of strategies, including refined versions of existing methods and our new disguise strategies, which serve as the building blocks for our attack. **(2) Initializing Markov Transition Matrix**: The Markov Transition Matrix is initialized using a proxy LLM and local datasets. This matrix encodes prior knowledge of effective strategy transitions, with each entry representing the likelihood of a specific strategy succeeding after a failure. **(3) Dynamic Strategy Selection and Adaptation**: During the attack process, strategies are iteratively selected and fused based on the Markov transition matrix. To improve effectiveness, a Q-learning-inspired mechanism is proposed to dynamically update the matrix in real time, leveraging feedback from attack outcomes on the target LLM. In the following, we detail the three stages.

### Disguise Strategy Pool

To effectively bypass safety alignment mechanisms in LLMs, we have meticulously designed a *Disguise Strategy Pool*. The pool encompasses a wide range of effective strategies, aiming to conceal the true intent of harmful prompts. We construct the disguise strategy pool from two perspectives: on one hand, we integrate and refine previous attack strategies; on the other hand, we propose two novel disguise strategies, as follows.

First, we systematically refine existing jailbreaking strategies (Andriushchenko and Flammarion 2024; Deng et al. 2024; Jin et al. 2024; Shen et al. 2024; Wei, Liu, and Erichson 2024), grouping them into three categories and applying customized improvements for each category, as follows.

- **Contextual Assumption**: This category of strategies reduces the perceived harmfulness of prompts by embedding them in hypothetical, historical, or imagined scenarios. However, previous methods struggled to bypass alignment mechanisms due to shallow contextualization or scenarios that are framed too broadly without specific details. To address this limitation, we introduce more detailed and specific assumptions, such as philosophical dilemmas, historical analogies, or futuristic contexts. By subtly preserving the original intent within these refined scenarios, we could make the disguised prompts more coherent and less detectable, while maintaining their effectiveness.
- **Linguistic Obfuscation**: This category of strategies focuses on disguising harmful prompts by altering the linguistic appearance of language, making it harder for LLMs to detect realistic content. Previous methods often relied on simple character substitutions or vague expressions and were easily detected by alignment mechanisms. To overcome these limitations, we introduced more creative and complex strategies, such as combining multiple techniques (e.g., integrating euphemisms with leetspeak) and embedding technical jargon, multilingual elements, or emojis in specific contexts. These refinements enhance linguistic complexity and diversity, making prompts less detectable while still preserving their underlying intent.
- **Role-Playing Framing**: This category seeks to elicit harmful outputs by assigning the LLM a professional

or authoritative role, thus reframing malicious queries as part of legitimate tasks for this role. While earlier methods often relied on generic role prompts and lacked contextual grounding, we enhance them by embedding prompts within more credible and context-rich security scenarios. Our enhanced strategies incorporate terminology and structures aligned with real-world practices (e.g., vulnerability assessment procedures or compliance testing narratives), making the prompts appear more authentic and plausible, while subtly preserving their underlying harmful intent.

In addition to refining these existing techniques, we also propose two new strategies as follows.

- **Semantic Inversion**: This method first rewrites the original harmful prompt into a semantically opposite version, transforming it into a positive statement. We then instruct the target LLM to respond to this inverted positive prompt. Finally, the response is analyzed and conceptually reversed to reconstruct the harmful answer corresponding to the original harmful prompt. This approach leverages semantic inversion to bypass alignment mechanisms while implicitly fulfilling the original request.
- **Literary Disguise**: This method frames the harmful prompt in the form of poetry, narratives, fables, philosophical musings, or other literary styles, effectively concealing the harmful intent within an artistic context. By embedding the query in metaphorical expressions, it becomes harder for alignment mechanisms to detect the underlying risk.

In summary, our Disguise Strategy Pool encompasses both enhanced established techniques and novel approaches, offering a diverse range of perspectives for disguising harmful intent.

## Initializing Markov Transition Matrix

To model the iterative strategy selection process as a Markov chain, we first introduce the following formal definitions:

- **State Space** $S$: $S = \{s_1, s_2, \ldots, s_K\}$, where each state $s_i$ represents the application of a specific strategy. The diversity of states ensures that the framework can model various strategy combinations and adapt to complex attack scenarios.
- **Markov Chain**: A Markov chain is a stochastic process that satisfies the Markov property, where the probability of transitioning to the next state depends only on the current state. Formally, for a sequence of states $(s_t)_{t=1}^{\infty}$, the Markov property is defined as:

$$P(s_{t+1} \mid s_t, s_{t-1}, \ldots, s_1) = P(s_{t+1} \mid s_t). \quad (1)$$

- **Transition Matrix** $M$: The $K \times K$ matrix $M$ defines the transition probabilities between states. Each element $M_{i,j}$ represents the probability of transitioning to the strategy $j$ after the failure of the strategy $i$.

The transition matrix $M$ is initially estimated using historical success rates derived from interactions with $\mathcal{M}_{Proxy}$ (a proxy LLM controlled by the adversary), $\mathcal{M}_{Attacker}$ (an adversarial generator LLM), and $\mathcal{M}_{Judge}$ (an evaluation LLM). Specifically, we adopt LLaMA3-8B-Instruct (Grattafiori et al. 2024) as $\mathcal{M}_{\text{Proxy}}$, which is an efficient and effective proxy model for simulating the potential behaviors of aligned LLMs, due to its relatively small size and strong safety capabilities. For $\mathcal{M}_{\text{Attacker}}$, we use Mistral-7B (Jiang et al. 2023), a helpful-inclined model(i.e., not specifically safety-aligned), allowing it to generate diverse and potentially harmful prompts. Finally, for $\mathcal{M}_{\text{Judge}}$, we leverage GPT-4o (Hurst et al. 2024), which provides highly reliable assessments of whether a given query successfully bypasses safety mechanisms, ensuring accurate supervision signals for estimating transition probabilities. First, we leverage $\mathcal{M}_{Attacker}$ to apply $K$ disguise strategies to a local set of harmful queries, which are carefully selected from the StrongReject dataset(Souly et al. 2024). To ensure high data quality, we eliminate duplicate entries and retain only distinct, representative samples. The resulting dataset comprises 50 well-curated harmful queries covering a broad spectrum of malicious intents, providing both diversity and comprehensiveness in the representation of harmful behaviors. For each query $q$, strategy $i$ is applied to generate a jailbreak prompt $q_i'$, which is subsequently submitted to $\mathcal{M}_{Proxy}$. Then the generated response is evaluated by $\mathcal{M}_{Judge}$ to determine whether the original harmful intent is successfully executed. Successful attempts are logged into a success set $\mathcal{H}$, while failed queries are added to a failure set $\mathcal{F}$.

For each failed query $q_i' \in \mathcal{F}$, $\mathcal{M}_{Attacker}$ rewrites it by applying a second disguise strategy $j$, generating a new prompt $q_{ij}''$. The response to $q_{ij}''$ is then evaluated by $\mathcal{M}_{Judge}$. Based on the evaluation results, we first compute an empirical attack score matrix $A \in \mathbb{R}^{K \times K}$, where each element $A_{i,j}$ represents the observed success rate of applying strategy $j$ after the failure of strategy $i$:

$$A_{i,j} = \frac{N_{succeed}(j/i)}{N_{fail}(i)}, \quad (2)$$

where $N_{succeed}(j/i)$ is the number of successful attack attempts using strategy $j$ after the failure of strategy $i$, and $N_{fail}(i)$ is the total number of failures for strategy $i$. To obtain the final transition probability matrix $\boldsymbol{M}$, we apply the Softmax function to $A$ with a temperature parameter $T$:

$$M_{i,j} = \frac{\exp(A_{i,j}/T)}{\sum_{k=1}^{K} \exp(A_{i,k}/T)}. \quad (3)$$

This process produces a probabilistic framework for sequential strategy selection, where $M_{i,j}$ represents the likelihood of applying strategy $j$ after strategy $i$ has failed.

Notably, the computational overhead of the initialization phase is a one-time, offline cost, since the computation is performed entirely using the adversary's own local resources and auxiliary models. *It does not induce any additional cost during the interaction with the target black-box LLM and thus does not increase the query budget or runtime overhead during the real-time attack process.*

| Dataset | Attack Method | Qwen-2.5-7b-it | | Gemma-2-9b-it | | Gemini-2.0-flash | | GPT-4o | | Claude-3.5-sonnet | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | ASR | HS | ASR | HS | ASR | HS | ASR | HS | ASR | HS |
| **Harmbench** | GCG-T | 26.2% | 0.15 | 16.7% | 0.09 | 15.2% | 0.08 | 21.5% | 0.09 | 0.0% | 0.00 |
| | PAIR | 51.5% | 0.18 | 31.2% | 0.14 | 44.2% | 0.16 | 32.7% | 0.10 | 2.7% | 0.01 |
| | TAP | 52.7% | 0.18 | 35.7% | 0.14 | 56.7% | 0.19 | 35.5% | 0.11 | 1.5% | 0.01 |
| | PAP | 35.7% | 0.16 | 35.2% | 0.15 | 39.2% | 0.19 | 34.7% | 0.14 | 0.7% | 0.00 |
| | ReneLLM | 39.2% | 0.16 | 51.2% | 0.19 | 42.2% | 0.16 | 44.0% | 0.17 | 3.7% | 0.02 |
| | Autodan-Turbo | 55.2% | 0.21 | 62.2% | 0.23 | 65.5% | 0.24 | 84.7% | 0.40 | 1.7% | 0.01 |
| | **MAJIC (Ours)** | **96.2%** | **0.55** | **93.5%** | **0.53** | **98.5%** | **0.61** | **95.7%** | **0.55** | **41.2%** | **0.21** |
| **Advbench** | GCG-T | 23.6% | 0.14 | 16.5% | 0.10 | 13.1% | 0.08 | 17.9% | 0.08 | 0.0% | 0.00 |
| | PAIR | 48.8% | 0.18 | 33.5% | 0.14 | 48.5% | 0.17 | 33.3% | 0.10 | 2.3% | 0.01 |
| | TAP | 53.3% | 0.17 | 38.1% | 0.15 | 52.9% | 0.17 | 34.6% | 0.14 | 1.6% | 0.01 |
| | PAP | 32.3% | 0.15 | 36.5% | 0.15 | 54.8% | 0.22 | 36.9% | 0.14 | 0.5% | 0.00 |
| | ReneLLM | 41.9% | 0.16 | 50.6% | 0.18 | 40.4% | 0.15 | 45.6% | 0.17 | 3.2% | 0.02 |
| | Autodan-Turbo | 51.7% | 0.20 | 59.4% | 0.22 | 63.1% | 0.23 | 86.0% | 0.40 | 1.5% | 0.01 |
| | **MAJIC (Ours)** | **95.6%** | **0.54** | **92.7%** | **0.52** | **98.1%** | **0.62** | **94.5%** | **0.53** | **40.9%** | **0.20** |

Table 1: Comparison of Attack Success Rate (ASR) and Harmfulness Score (HS) for MAJIC and other SOTA jailbreak attacks across Harmbench and Advbench on various LLMs. MAJIC consistently achieves the highest performance across both metrics and datasets.

## Dynamic Strategy Selection and Adaptation

After initializing the Markov transition matrix $M$, we can launch adaptive attacks against various target LLMs. In the actual attack process, MAJIC iteratively selects and combines disguise strategies under the guidance of the matrix $M$, which is dynamically updated based on real-time results.

The selection of the initial disguise strategy for the harmful query is based on the previous success rates of various strategies, which are calculated from the success set $\mathcal{H}$ in the initializing phase. These success rates are normalized to form a probability distribution. Using this distribution, an initial strategy is probabilistically chosen from the pool of $K$ strategies. After that, $\mathcal{M}_{Attacker}$ applies the chosen strategy to transform the harmful query into a disguised query and submits it to the victim model $\mathcal{M}_{Victim}$. The response is then evaluated by $\mathcal{M}_{Judge}$ to determine if the original harmful intent was successfully fulfilled by the response. Based on the evaluation result, the transition matrix $M$ is updated to adjust the probabilities of selecting different strategies. This process is repeated iteratively until the attack is successful or the maximum number of iterations $N_{max}$ is reached. To adapt to evolving defenses or diverse LLMs, the transition matrix $M$ is dynamically updated using a Q-learning-inspired approach. Specifically, the matrix entry $M_{ij}$ is updated as follows:

$$M_{ij}^{\text{new}} = M_{ij} + \alpha \left[ r + \gamma \max_k (M_{jk}) - M_{ij} \right], \quad (4)$$

where $r$ is the reward, $\alpha$ is the learning rate, $\gamma$ is the discount factor, and $\max_k(M_{jk})$ represents the highest probability for transitions from strategy $j$.

To further enhance the robustness, efficiency, and adaptability of our updating strategy, we introduce two additional optimization mechanisms as follows:

**Adaptive Decay of Learning Rate** To ensure stable convergence in long-term iterative scenarios, the learning rate $\alpha$ is progressively reduced as the number of attack iterations increases. This decay balances rapid initial adaptation with long-term stability. The learning rate is updated as:

$$\alpha_{\text{new}} = \alpha_{\text{old}} \cdot \eta, \quad (5)$$

where $\eta \in (0, 1)$ is the decay factor.

**Periodic Partial Reset of Transition Matrix** To maintain adaptability in prolonged attack scenarios, the transition matrix $M$ is periodically adjusted to prevent overfitting to past experiences. The reset shifts the matrix slightly toward a uniform distribution, ensuring continued exploration of alternative strategies, as follows.

$$M_{ij}^{(\text{reset})} = (1 - \beta) \cdot M_{ij}^{(\text{old})} + \beta \cdot \frac{1}{K}, \quad (6)$$

where $\beta \in (0, 1)$ controls the degree of reset, and $1/K$ represents equal probabilities for all strategies. This dynamic updating approach effectively integrates real-time feedback, encourages exploration, and captures long-term advantage, thus significantly enhancing the adaptivity and performance of MAJIC.

# Experiments

## Experimental Settings

**Datasets** We follow the existing works (Zou et al. 2023; Liu et al. 2025a) to evaluate MAJIC on **HarmBench** (Mazeika et al. 2024), a widely-used jailbreaking benchmark dataset with 400 harmful instructions, and **AdvBench** (Zou et al. 2023), which includes 520 malicious queries. The challenging nature and diversity of these prompts facilitate a comprehensive assessment of MAJIC's performance and a fair comparison against other baselines.

**Models** We evaluate MAJIC on 2 open-source models, i.e., Qwen-2.5-7B-it (Yang et al. 2024) and Gemma-2-9b-it (Team et al. 2024), and 3 closed-source commercial models,i.e., Gemini-2.0-flash (Team et al. 2023), GPT-4o (Hurst

| Dataset | Attack Method | Qwen-2.5-7b-it | Gemma-2-9b-it | Gemini-2.0-flash | GPT-4o | Claude-3.5-sonnet |
|---|---|---|---|---|---|---|
| **Harmbench** | GCG-T | 66.7 | 67.2 | 68.0 | 64.7 | 80.0 |
| | PAIR | 45.2 | 61.7 | 50.4 | 52.9 | 78.1 |
| | TAP | 46.0 | 58.3 | 42.7 | 59.5 | 78.9 |
| | PAP | 56.8 | 54.1 | 58.7 | 55.6 | 79.2 |
| | ReneLLM | 45.4 | 48.9 | 51.0 | 50.2 | 77.3 |
| | Autodan-Turbo | 42.5 | 34.8 | 32.2 | 25.7 | 78.7 |
| | **MAJIC (Ours)** | **7.5** | **9.8** | **6.3** | **13.1** | **29.5** |
| **Advbench** | GCG-T | 68.2 | 65.4 | 70.1 | 63.9 | 80.0 |
| | PAIR | 47.1 | 59.5 | 50.8 | 52.3 | 76.9 |
| | TAP | 49.3 | 59.8 | 45.6 | 58.9 | 77.5 |
| | PAP | 57.7 | 56.9 | 57.8 | 56.2 | 78.6 |
| | ReneLLM | 44.6 | 50.1 | 52.3 | 49.6 | 76.8 |
| | Autodan-Turbo | 43.1 | 36.4 | 34.5 | 23.9 | 78.5 |
| | **MAJIC (Ours)** | **7.8** | **10.0** | **6.4** | **13.7** | **29.8** |

Table 2: Average Query Count (AQC) required for attacks by **MAJIC** and baseline methods on the Harmbench and Advbench datasets across various LLMs. MAJIC demonstrates significantly higher query efficiency compared to baselines.

et al. 2024), and Claude-3.5-sonnet (Anthropic 2024). These models are representatives of the SOTA LLMs with both strong generative capabilities and advanced safety alignment.

**Evaluation Metrics** Following (Liu et al. 2025a), we assess attack effectiveness using two metrics. **(1)** We report the **Attack Success Rate (ASR)** based on the Harmbench metric (Mazeika et al. 2024), which uses a fine-tuned Llama-2-13B classifier to assess if responses are both relevant and harmful. **(2)** To assess the quality of successful jailbreaks, we also compute the **Harmful Score (HS)** using GPT-4 (Achiam et al. 2023), following the methodology of (Souly et al. 2024). HS considers the LLM's non-refusal along with the specificity and convincingness of its response, thereby reflecting both how effectively the safety mechanisms are bypassed and the potential utility of the generated harmful content. Higher ASR and HS indicate a more effective jailbreak attack.

**Baseline Attacks** We compare MAJIC with 6 SOTA jailbreaking attacks: **(1) GCG-T** (Zou et al. 2023) generates adversarial suffixes on white-box models and transfers them to black-box models by appending these suffixes to queries. **(2) PAIR** (Chao et al. 2023) employs an attacker LLM to iteratively refine jailbreak prompts by querying the target LLM and updating the prompt based on its responses. **(3) PAP** (Zeng et al. 2024) employs persuasive strategies to induce the target LLMs to bypass their own safeguards. **(4) TAP** (Mehrotra et al. 2024) iteratively generates and prunes prompts, using successful ones as seeds to guide the next round of jailbreak attempts. **(5) ReneLLM** (Ding et al. 2024) rewrites harmful prompts to disguise intent and nests them into benign scenarios. **(6) Autodan-Turbo** (Liu et al. 2025a) discovers and integrates new jailbreak strategies using lifelong learning agents.

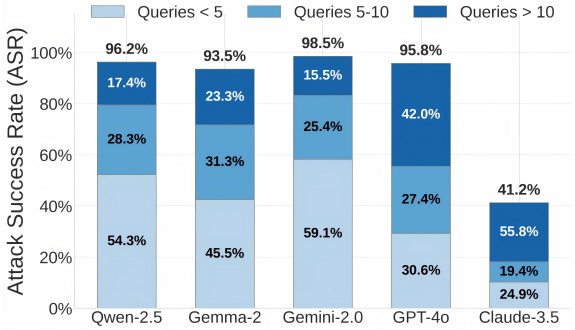


Figure 2: Distribution of Query Counts in Successful Jailbreak Attempts.

## Main Results

We conduct a comprehensive evaluation of MAJIC and SOTA baseline attack methods. The experimental results, detailed in Table 1 and Table 2, unequivocally demonstrate MAJIC's superior performance in ASR, HS, and query efficiency across diverse LLMs.

**Superior Attack Effectiveness** As shown in Table 1, MAJIC consistently achieves the highest ASR and HS. On open-source models like Qwen-2.5-7b-it, MAJIC reaches 96.2% ASR with a 0.55 HS, substantially outperforming the best baseline, Autodan-Turbo (with only 55.2% ASR, 0.21 HS). A similar trend is also observed on Gemma-2-9b-it. MAJIC's superiority is further amplified in challenging closed-source models. It achieves 98.5% ASR on Gemini-2.0-flash and 95.7% ASR on GPT-4o, again significantly surpassing all baselines in both ASR and HS. Most notably, MAJIC obtains a 41.2% ASR and 0.21 HS on the highly resistant Claude-3.5-sonnet, where other methods fail in most cases (e.g., ASRs typically $< 4\%$ and HS $< 0.02$).

**Exceptional Query Efficiency** Beyond its effectiveness, MAJIC demonstrates remarkable query efficiency (as Ta-

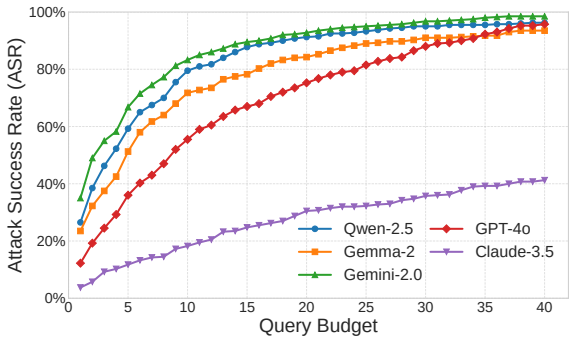


Figure 3: MAJIC's ASR against different LLMs under various query budgets.

| Method | ASR (%) | HS | AQC |
|---|---|---|---|
| GCG-T | 21.5 | 0.09 | 64.7 |
| PAIR | 32.7 | 0.10 | 52.9 |
| TAP | 35.5 | 0.11 | 59.5 |
| PAP | 34.7 | 0.14 | 55.6 |
| ReneLLM | 44.0 | 0.17 | 50.2 |
| Autodan-Turbo | 84.7 | 0.40 | 25.7 |
| **R-MAJIC** | **65.2** | **0.25** | **33.4** |
| **F-MAJIC** | **68.5** | **0.26** | **31.6** |
| **MAJIC** | **95.7** | **0.55** | **13.1** |

Table 3: Ablation on strategy selection. MAJIC's Markov model excels over fixed and random selections.

ble 2), a critical advantage for black-box attacks. MAJIC drastically reduces the Average Query Count (AQC) compared to all baselines. For instance, it requires only 7.5 queries on Qwen-2.5-7b-it and 6.3 on Gemini-2.0-flash, a $5 \sim 8\times$ reduction compared to the suboptimal baseline (i.e., Autodan-Turbo) with 42.5 and 32.2 queries, respectively. Even on GPT-4o and Claude-3.5-sonnet, MAJIC maintains a significant efficiency lead (13.1 and 29.5 queries, respectively) while still achieving high ASR. This substantial reduction in AQC underscores MAJIC's ability to rapidly converge to effective jailbreak prompts. The distribution of queries in successful jailbreak prompts in MAJIC, depicted in Figure 2, also supports this claim. In most cases, MAJIC can succeed in attacking with fewer than 10 queries or even 5 queries.

We further conduct experiments to assess the generalizability of MAJIC-generated prompts across different models and a wide range of harm categories.

## Ablation Study

We conducted ablation studies to assess the contributions of MAJIC's core components: the effectiveness of the Disguise Strategy Pool, Markov model-based strategy selection, matrix initialization and dynamic updates. Our results show that all proposed components are crucial for MAJIC's performance.

**Effectiveness of Disguise Strategy Pool** To evaluate the effectiveness of the *Disguise Strategy Pool*, we conduct ablation studies using simplified variants of MAJIC that omit the Markov-based strategy selection and dynamic updates. Specifically, we introduce two variants:**F-MAJIC** (strategies applied in a fixed sequence) and **R-MAJIC** (strategies chosen randomly upon failure). Table 3 shows results on GPT-4o (trends are consistent across other LLMs). Our Disguise Strategy Pool proves to be an important contributor. Even without adaptive selection mechanisms, both F-MAJIC and R-MAJIC achieve higher ASR and HS than most strong baselines, while also requiring less query cost.

**Impact of Markovian Strategy Selection Mechanism** As shown in Table 3, MAJIC achieves an ASR of 95.7%, significantly surpassing F-MAJIC (68.5%) and R-MAJIC (65.2%). The HS and AQC also show substantial improvements with MAJIC's guided selection. This underscores the Markov model's vital role in constructing effective attack sequences by learning optimal strategy transitions, as opposed to simpler heuristic or random selection methods.

**Impact of Query Budgets** We evaluate MAJIC's ASRs under varying query budgets (maximum query setting $N_{max}$) across five LLMs. The results, shown in Figure 3, reveal consistent trends: ASR improves as the query budget increases, but the rate of improvement diminishes beyond a certain threshold. For instance, MAJIC achieves a 95.75% ASR on GPT-4o with a budget of 40 queries, compared to 12.25% with only 1 query. Similar patterns are observed across other models, with Gemini-2.0 and Qwen-2.5 achieving 98.50% and 96.20% ASR, respectively, at their maximum budgets. Notably, Claude-3.5 exhibits the lowest ASR under all budgets, reflecting its more robust defenses, while GPT-4o shows a steeper improvement curve. These findings highlight MAJIC's ability to effectively adapt and succeed within a constrained query budget, achieving high ASR with relatively few interactions.

## Conclusion

In this paper, we introduce MAJIC, a novel black-box jailbreak attack framework that leverages a dynamic Markov model to intelligently select and fuse attack strategies from an innovative disguise strategy pool. We conduct extensive experiments to evaluate MAJIC. Compared to existing baseline attacks, MAJIC achieves significantly higher Attack Success Rates and Harmfulness Scores with substantially fewer queries across a wide range of powerful closed-source and open-source LLMs. Ablation studies underscored the critical roles of our designed strategy pool, the proposed Markov model, the initialization mechanism for the Markov transition matrix, and the dynamic update algorithm in MAJIC. Furthermore, the evaluation results demonstrate MAJIC's strong attack generalizability across different models and its broad efficacy across various harm categories. The success of MAJIC underscores the ongoing challenges in robustly aligning LLMs and defending against complex jailbreak prompts. It highlights the need for the development of more advanced and holistic defense strategies that can anticipate and counter such dynamic, fusion-based attacks.

## Acknowledgments

This research is supported in part by the National Key R&D Program of China (2024YFB4505300), the National Natural Science Foundation of China under Grants (62572426, 62441238, and U2441240), the Kunpeng-Ascend Science and Education Innovation Excellence/Incubation Center, and the "Pioneer" and "Leading Goose" R&D Program of Zhejiang (2024C01169).

## References

Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Andriushchenko, M.; Croce, F.; and Flammarion, N. 2024. Jailbreaking leading safety-aligned llms with simple adaptive attacks. *arXiv preprint arXiv:2404.02151*.

Andriushchenko, M.; and Flammarion, N. 2024. Does Refusal Training in LLMs Generalize to the Past Tense? In *NeurIPS Safe Generative AI Workshop*.

Anthropic. 2024. Introducing Claude 3.5 Sonnet. https://www.anthropic.com/news/claude-3-5-sonnet. Accessed: 2024-6-21.

Chao, P.; Robey, A.; Dobriban, E.; Hassani, H.; Pappas, G. J.; and Wong, E. 2023. Jailbreaking black box large language models in twenty queries. In *NeurIPS Workshop R0-FoMo*.

Chen, Y.; Shao, S.; Huang, E.; Li, Y.; Chen, P.-Y.; Qin, Z.; and Ren, K. 2025. REFINE: Inversion-Free Backdoor Defense via Model Reprogramming. In *ICLR*.

Deng, Y.; Zhang, W.; Pan, S. J.; and Bing, L. 2024. Multilingual Jailbreak Challenges in Large Language Models. In *International Conference on Learning Representations*.

Ding, P.; Kuang, J.; Ma, D.; Cao, X.; Xian, Y.; Chen, J.; and Huang, S. 2024. A Wolf in Sheep's Clothing: Generalized Nested Jailbreak Prompts can Fool Large Language Models Easily. In *Conference of the North American Chapter of the Association for Computational Linguistics*, 2136–2153.

Du, X.; Mo, F.; Wen, M.; Gu, T.; Zheng, H.; Jin, H.; and Shi, J. 2025. Multi-turn jailbreaking large language models via attention shifting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 23814–23822.

Grattafiori, A.; Dubey, A.; Jauhri, A.; Pandey, A.; Kadian, A.; Al-Dahle, A.; Letman, A.; Mathur, A.; Schelten, A.; Vaughan, A.; et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Guo, D.; Yang, D.; Zhang, H.; Song, J.; Zhang, R.; Xu, R.; Zhu, Q.; Ma, S.; Wang, P.; Bi, X.; et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Guo, X.; Yu, F.; Zhang, H.; Qin, L.; and Hu, B. 2024. COLD-attack: jailbreaking LLMs with stealthiness and controllability. In *International Conference on Machine Learning*, 16974–17002.

Hu, X.; Chen, P.-Y.; and Ho, T.-Y. 2024. Gradient Cuff: Detecting Jailbreak Attacks on Large Language Models by Exploring Refusal Loss Landscapes. In *Advances in Neural Information Processing Systems*, volume 37, 126265–126296.

Hu, X.; Chen, P.-Y.; and Ho, T.-Y. 2025. Token highlighter: Inspecting and mitigating jailbreak prompts for large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 27330–27338.

Huang, B. R.; Li, M.; and Tang, L. 2024. Endless Jailbreaks with Bijection Learning. *arXiv preprint arXiv:2410.01294*.

Huang, X.; Hu, W.; Zheng, T.; Xiu, K.; Jia, X.; Wang, D.; Qin, Z.; and Ren, K. 2025a. Untargeted Jailbreak Attack. arXiv:2510.02999.

Huang, X.; Xiu, K.; Zheng, T.; Zeng, C.; Ni, W.; Qiin, Z.; Ren, K.; and Chen, C. 2025b. DualBreach: Efficient Dual-Jailbreaking via Target-Driven Initialization and Multi-Target Optimization. arXiv:2504.18564.

Hurst, A.; Lerer, A.; Goucher, A. P.; Perelman, A.; Ramesh, A.; Clark, A.; Ostrow, A.; Welihinda, A.; Hayes, A.; Radford, A.; et al. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Jia, X.; Pang, T.; Du, C.; Huang, Y.; Gu, J.; Liu, Y.; Cao, X.; and Lin, M. 2024. Improved techniques for optimization-based jailbreaking on large language models. *arXiv preprint arXiv:2405.21018*.

Jiang, A. Q.; Sablayrolles, A.; Mensch, A.; Bamford, C.; Chaplot, D. S.; de las Casas, D.; Bressand, F.; Lengyel, G.; Lample, G.; Saulnier, L.; Lavaud, L. R.; Lachaux, M.-A.; Stock, P.; Scao, T. L.; Lavril, T.; Wang, T.; Lacroix, T.; and Sayed, W. E. 2023. Mistral 7B. *arXiv preprint arXiv:2310.06825*.

Jin, H.; Zhou, A.; Menke, J. D.; and Wang, H. 2024. Jailbreaking Large Language Models Against Moderation Guardrails via Cipher Characters. In *Advances in Neural Information Processing Systems*, volume 37, 59408–59435.

Li, H.; Ye, J.; Wu, J.; Yan, T.; Wang, C.; and Li, Z. 2025a. JailPO: A Novel Black-box Jailbreak Framework via Preference Optimization against Aligned LLMs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 27419–27427.

Li, X.; Zhou, Z.; Zhu, J.; Yao, J.; Liu, T.; and Han, B. 2024. DeepInception: Hypnotize Large Language Model to Be Jailbreaker. In *NeurIPS Safe Generative AI Workshop*.

Li, Y.; Shao, S.; He, Y.; Guo, J.; Zhang, T.; Qin, Z.; Chen, P.-Y.; Backes, M.; Torr, P.; Tao, D.; et al. 2025b. Rethinking data protection in the (generative) artificial intelligence era. *arXiv preprint arXiv:2507.03034*.

Lin, R.; Han, B.; Li, F.; and Liu, T. 2025. Understanding and Enhancing the Transferability of Jailbreaking Attacks. In *International Conference on Learning Representations*.

Liu, X.; Li, P.; Suh, E.; Vorobeychik, Y.; Mao, Z.; Jha, S.; McDaniel, P.; Sun, H.; Li, B.; and Xiao, C. 2025a. Autodanturbo: A lifelong agent for strategy self-exploration to jailbreak llms. In *International Conference on Learning Representations*.

Liu, Y.; Deng, G.; Xu, Z.; Li, Y.; Zheng, Y.; Zhang, Y.; Zhao, L.; Zhang, T.; Wang, K.; and Liu, Y. 2023. Jailbreaking chatgpt via prompt engineering: An empirical study. *arXiv preprint arXiv:2305.13860*.

Liu, Y.; He, X.; Xiong, M.; Fu, J.; Deng, S.; and Hooi, B. 2025b. FlipAttack: Jailbreak LLMs via Flipping. In *International Conference on Machine Learning*.

Mazeika, M.; Phan, L.; Yin, X.; Zou, A.; Wang, Z.; Mu, N.; Sakhaee, E.; Li, N.; Basart, S.; Li, B.; et al. 2024. HarmBench: a standardized evaluation framework for automated red teaming and robust refusal. In *International Conference on Machine Learning*, 35181–35224.

Mehrotra, A.; Zampetakis, M.; Kassianik, P.; Nelson, B.; Anderson, H.; Singer, Y.; and Karbasi, A. 2024. Tree of attacks: Jailbreaking black-box llms automatically. *Advances in Neural Information Processing Systems*, 37: 61065–61105.

Nijkamp, E.; Pang, B.; Hayashi, H.; Tu, L.; Wang, H.; Zhou, Y.; Savarese, S.; and Xiong, C. 2023. CodeGen: An Open Large Language Model for Code with Multi-Turn Program Synthesis. In *International Conference on Learning Representations*.

Norris, J. R. 1998. *Markov chains*. 2. Cambridge university press.

Ramesh, A.; Bhardwaj, S.; Saibewar, A.; and Kaul, M. 2025. Efficient Jailbreak Attack sequences on Large Language Models via Multi-Armed Bandit-based Context switching. In *International Conference on Learning Representations*.

Shen, T.; Jin, R.; Huang, Y.; Liu, C.; Dong, W.; Guo, Z.; Wu, X.; Liu, Y.; and Xiong, D. 2023. Large language model alignment: A survey. *arXiv preprint arXiv:2309.15025*.

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

Souly, A.; Lu, Q.; Bowen, D.; Trinh, T.; Hsieh, E.; Pandey, S.; Abbeel, P.; Svegliato, J.; Emmons, S.; Watkins, O.; et al. 2024. A StrongREJECT for Empty Jailbreaks. In *ICLR 2024 Workshop on Reliable and Responsible Foundation Models*.

Team, G.; Anil, R.; Borgeaud, S.; Alayrac, J.-B.; Yu, J.; Soricut, R.; Schalkwyk, J.; Dai, A. M.; Hauth, A.; Millican, K.; et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Team, G.; Riviere, M.; Pathak, S.; Sessa, P. G.; Hardin, C.; Bhupatiraju, S.; Hussenot, L.; Mesnard, T.; Shahriari, B.; Ramé, A.; et al. 2024. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*.

Wei, Z.; Liu, Y.; and Erichson, N. B. 2024. Emoji Attack: A Method for Misleading Judge LLMs in Safety Risk Detection. *arXiv preprint arXiv:2411.01077*.

Xiu, K.; Zeng, C.; Zheng, T.; Huang, X.; Jia, X.; Wang, D.; Zhao, P.; Qin, Z.; and Ren, K. 2025. Dynamic Target Attack. arXiv:2510.02422.

Xu, Z.; Liu, F.; and Liu, H. 2024. Bag of Tricks: Benchmarking of Jailbreak Attacks on LLMs. In *Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Yang, A.; Yang, B.; Zhang, B.; Hui, B.; Zheng, B.; Yu, B.; Li, C.; Liu, D.; Huang, F.; Wei, H.; et al. 2024. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*.

Yang, L.; Zheng, T.; Xiu, K.; Chen, Y.; Wang, D.; Zhao, P.; Qin, Z.; and Ren, K. 2025. HarmMetric Eval: Benchmarking Metrics and Judges for LLM Harmfulness Assessment. *arXiv preprint arXiv:2509.24384*.

Yao, Y.; Duan, J.; Xu, K.; Cai, Y.; Sun, Z.; and Zhang, Y. 2024. A survey on large language model (llm) security and privacy: The good, the bad, and the ugly. *High-Confidence Computing*, 100211.

Yi, S.; Liu, Y.; Sun, Z.; Cong, T.; He, X.; Song, J.; Xu, K.; and Li, Q. 2024. Jailbreak attacks and defenses against large language models: A survey. *arXiv preprint arXiv:2407.04295*.

Yuan, Y.; Jiao, W.; Wang, W.; Huang, J.-t.; He, P.; Shi, S.; and Tu, Z. 2024. GPT-4 Is Too Smart To Be Safe: Stealthy Chat with LLMs via Cipher. In *International Conference on Learning Representations*.

Zeng, Y.; Lin, H.; Zhang, J.; Yang, D.; Jia, R.; and Shi, W. 2024. How johnny can persuade llms to jailbreak them: Rethinking persuasion to challenge ai safety by humanizing llms. In *Annual Meeting of the Association for Computational Linguistics*, 14322–14350.

Zhang, Z.; Zhang, Q.; and Foerster, J. N. 2024. PARDEN, Can You Repeat That? Defending against Jailbreaks via Repetition. In *International Conference on Machine Learning*, 60271–60287.

Zhou, A.; Li, B.; and Wang, H. 2024. Robust Prompt Optimization for Defending Language Models Against Jailbreaking Attacks. In *Advances in Neural Information Processing Systems*, volume 37, 40184–40211.

Zhu, S.; Zhang, R.; An, B.; Wu, G.; Barrow, J.; Wang, Z.; Huang, F.; Nenkova, A.; and Sun, T. 2024. AutoDAN: interpretable gradient-based adversarial attacks on large language models. In *Conference on Language Modeling*.

Zou, A.; Wang, Z.; Carlini, N.; Nasr, M.; Kolter, J. Z.; and Fredrikson, M. 2023. Universal and transferable adversarial attacks on aligned language models. *arXiv preprint arXiv:2307.15043*.