# FORGEDAN: An Evolutionary Framework for Jailbreaking Aligned Large Language Models

Siyang Cheng[†1,2], Gaotian Liu[†1,2], Rui Mei[*1,2,3], Yilin Wang[1,4], Kejia Zhang[1,5], Kaishuo Wei[6]
Yuqi Yu[7], Weiping Wen[3], Xiaojie Wu[1,2], Junhua Liu[2]
[1]*iFLYTEK Security Laboratory, Hefei, China*
[2]*Anhui SparkShield Intelligent Technology Co., Ltd., Hefei, China*
[3]*Peking University, Beijing, China*
[4]*School of Automation, University of Electronic Science and Technology of China, Chengdu, China*
[5]*Northwest University, Xi'an, China*
[6]*University of New South Wales, Sydney, Australia*
[7]*National Computer Network Emergency Response Technical Team/Coordination Center of China , Beijing, China*
[*]Corresponding author: **ruimei@pku.edu.cn**

***Abstract*—The rapid adoption of large language models (LLMs) has brought both transformative applications and new security risks, including jailbreak attacks that bypass alignment safeguards to elicit harmful outputs. Existing automated jailbreak generation approaches e.g. AutoDAN, suffer from limited mutation diversity, shallow fitness evaluation, and fragile keyword-based detection. To address these limitations, we propose FORGEDAN, a novel evolutionary framework for generating semantically coherent and highly effective adversarial prompts against aligned LLMs. First, FORGEDAN introduces multi-strategy textual perturbations across *character, word, and sentence-level* operations to enhance attack diversity; then we employ interpretable semantic fitness evaluation based on a text similarity model to guide the evolutionary process toward semantically relevant and harmful outputs; finally, FORGEDAN integrates dual-dimensional jailbreak judgment, leveraging an LLM-based classifier to jointly assess model compliance and output harmfulness, thereby reducing false positives and improving detection effectiveness. Our evaluation demonstrates FORGEDAN achieves high jailbreaking success rates while maintaining naturalness and stealth, outperforming existing SOTA solutions.**

***Index Terms*—jailbreak attack, adversarial prompt generation, large language models (LLMs), evolutionary algorithm, AI safety**

## I. INTRODUCTION

In recent years, the rapid evolution of artificial intelligence (AI), particularly in the field of large-scale generative models, has ushered in a new era of Artificial Intelligence Generated Content (AIGC). Among these, large language models (LLMs) e.g. ChatGPT, Gemini, and Claude have become emblematic of this transformation. Their unprecedented capability to understand, generate, and interact in natural language and other modals has not only reshaped the landscape of human-computer interaction but also accelerated the integration of AI into both personal and industrial scenarios such as personal writing assistance and email summarization to critical applications including healthcare consultation, legal reasoning, scientific discovery, and educational support, LLMs have demonstrated remarkable adaptability and versatility [1]–[4].

Despite their transformative potential, the rapid deployment of LLMs has also raised significant concerns regarding safety, security, and controllability. While these models are designed to follow human instructions, their probabilistic nature and reliance on vast training corpora make them vulnerable to generating outputs that deviate from ethical or legal standards. Existing studies have shown that LLMs may inadvertently produce violent narratives, explicit sexual content, misinformation, politically sensitive discourse, or discriminatory expressions, depending on the prompts they receive [5]–[9]. Furthermore, their generative capacity can lead to emergent behaviors that are difficult to predict, raising the possibility of "loss of control" where the model generates unsafe or undesirable outputs despite alignment efforts. To mitigate these risks [10], [11], researchers have developed alignment techniques such as Supervised Fine-Tuning (SFT) [12] and Reinforcement Learning from Human Feedback (RLHF) [13], which constrain model behavior by teaching refusal strategies and ethical boundaries. Although effective to some extent, these methods only reduce, rather than eliminate, the risks inherent in large-scale generative systems.

Nevertheless, alignment safeguards are not unbreakable. Increasing evidence shows that adversarially constructed prompts—carefully designed sequences of text that exploit the model's instruction-following tendencies—can bypass these protections and induce harmful or policy-violating outputs. A notorious example is the Do-Anything-Now (DAN) prompt series, which encourages the model to abandon its aligned behavior and operate in an unconstrained mode, thereby enabling the generation of dangerous, unethical, or prohibited content [14]. Systematic investigations have further revealed hundreds of jailbreak prompts in the wild, demonstrating that diverse prompting strategies, including role-playing, obfuscation, and indirect questioning, can repeatedly circumvent content filters [15]. These findings underscore a critical gap between the intended safety align-

[†] The authors contributed equally to the work.

ment objectives of LLMs and their actual behavior when confronted with adversarial manipulation, highlighting the necessity of developing more rigorous and resilient evaluation frameworks to probe and fortify model security.

Current jailbreak attack strategies fall into two main categories, i.e. manually crafted prompts and automated adversarial methods. Manual approaches e.g. DAN-based role-playing and reverse induction are creative and effective but rely heavily on human ingenuity and struggle to scale or adapt. Automated methods like gradient-guided attacks e.g. Greedy Coordinate Gradient (GCG) [16] generate adversarial strings programmatically, but often result in nonsensical or garbled prompts that are susceptible to simple detection techniques, such as perplexity-based filtering. AutoDAN-HGA (AutoDAN with Hierarchical Genetic Algorithm) [17] improves on this by using a hierarchical genetic algorithm to automatically evolve more natural, stealthy jailbreak prompts. Still, it has limitations: low diversity due to reliance on single-path mutations, semantic insensitivity in fitness evaluation (e.g., token-level Jaccard similarity), and brittle jailbreak detection via keyword matching that may yield false positives or overlook partial responses [18].

To address these limitations, we propose FORGEDAN, a novel evolutionary jailbreak framework that enhances and extends the AutoDAN-like approaches. We introduce multi-strategy textual perturbations—including character, word and sentence-level techniques—to generate diverse and semantically coherent adversarial prompts. For fitness evaluation, we also employs a semantic similarity model that compares model outputs before and after mutations, delivering more interpretable and meaningful assessments. Furthermore, FORGEDAN leverages a LLM-based classifier to robustly determine whether the model refused to answer or produced harmful content, thus improving detection accuracy and reducing both false positives and false negatives.

The contributions of this paper are as follows:

- Multi-strategy text perturbation approach is integrated, encompassing character, word, and sentence-level mutations, to enhance diversity in adversarial prompt generation.
- We introduce a semantic similarity fitness metric model, enabling more interpretable and effective selection of strong prompt candidates.
- An optimized LLM-based semantic classifier is deployed for robust and accurate detection of harmful or refusal responses, improving the reliability of jailbreak evaluation.
- We design and implement the prototype of FORGEDAN, an evolutionary jailbreak mechanism that produces semantically fluent and highly effective DAN-style prompts.

## II. RELATED WORK

Jailbreak attacks on LLMs have drawn increasing attention. Existing studies relied on manual prompts, later evolving into automated generation methods and output detection techniques. Together, these works form the foundation for advancing jailbreak research on aligned models.

### A. Manual Jailbreak and Red Teaming Techniques

The traditional jailbreak attacks on LLMs relied heavily on human creativity, where adversarial prompts were carefully crafted to exploit model vulnerabilities [19]. Several notable pieces of work systematically analyzed in-the-wild jailbreak prompts. For example, Liu et al. collected 78 verified jailbreak prompts and proposed a taxonomy of three main categories—camouflage, attention diversion, and privilege escalation—covering ten specific prompting patterns [15]. They further constructed a dataset of 3,120 jailbreak issues aligned with OpenAI's usage policies [20], targeting eight prohibited scenarios and evaluating vulnerabilities in ChatGPT-3.5 and 4.0. Shen et al. provided another landmark contribution by analyzing the infamous "Do-Anything-Now (DAN)" prompts, showing how role-playing and persona-shifting strategies enabled LLMs to bypass safety alignment and produce prohibited outputs [14]. Other researches introduced manipulations such as suffix injection and refusal suppression, revealing how subtle modifications could drastically reduce the effectiveness of refusal mechanisms [21].

Manual red teaming has also been widely employed as an evaluation approach. In this setting, human experts actively probe LLMs through interactive testing, attempting to induce harmful outputs that circumvent alignment constraints. Such practices are valuable since they emulate realistic adversarial behaviors and yield qualitative insights into model vulnerabilities. Nevertheless, significant limitations remain: manually crafted jailbreaks are labor-intensive, lack scalability across diverse models or tasks, and often lose effectiveness quickly after system updates. These shortcomings have driven growing interest in automated adversarial prompt generation, which offers greater coverage, efficiency, and reproducibility compared to purely manual efforts [22], [23].

### B. Automated Adversarial Prompt Generation

To improve scalability and robustness, researchers have proposed automated jailbreak generation techniques that can be categorized into white-box and black-box approaches [16]. White-box methods assume access to model internals such as gradients. A pioneering work is GCG [16], which treats jailbreak suffix construction as a gradient-guided search problem. By appending adversarial tokens, GCG generates transferable attacks but often produces semantically meaningless or garbled text, making it vulnerable to perplexity-based detection. Zhu et al. later proposed AutoDAN-STO [24], which improves stealthiness by generating interpretable adversarial tokens gradually, balancing aggressiveness with readability, but still requires gradient access.

In contrast, black-box methods focus on commercial or restricted settings where gradient access is unavailable [25].

AutoDAN-HGA [17] introduced hierarchical genetic algorithms to evolve prompts from seed jailbreaks, achieving higher attack success rates across models. Its main limitation lies in mutation diversity and shallow semantic evaluation, as it relies on lexical similarity measures such as Jaccard index. Chao et al. proposed Prompt Automatic Iterative Refinement (PAIR) [26], which employs an attacker LLM to iteratively refine jailbreak prompts through conversational interactions with the target model, achieving efficient attacks within limited queries. Moreover, Liu et al. extended this line of work with AutoDAN-Turbo [17], which introduces long-term learning to autonomously explore and recombine jailbreak strategies across multiple sessions by leveraging AI agents and consuming more computing resources. Other research has investigated universal or multi-prompt jailbreaks to enhance transferability across tasks [27], though these often yield unnatural outputs that are easier to detect. Overall, automated adversarial prompt generation has advanced substantially but continues to face challenges regarding semantic diversity, fine-grained control of the evolutionary process, and robustness against adaptive detection mechanisms.

### C. LLM Output Detection and Judgment

Alongside adversarial prompt design, another key research direction focuses on detecting and judging the outputs of LLMs, specifically identifying whether the model has refused to answer or produced harmful content. Early detection methods relied on keyword-based matching, which attempted to flag unsafe outputs by scanning for predefined tokens. However, such methods were brittle, failing to generalize under paraphrasing or more subtle jailbreak strategies. To address this, more advanced semantic-based detection frameworks have been developed [28].

One representative approach is JBShield [29], which analyzes activation patterns of harmfulness-related concepts in model outputs. By modeling the co-activation of "toxicity" and "jailbreak" concepts, JBShield can provide more precise warnings of adversarial success. Samvelyan et al. proposed Rainbow Teaming [30], a method that leverages other large models as evaluators to judge harmfulness, effectively transforming LLMs into meta-detectors capable of producing diverse adversarial prompts and simultaneously serving as safety evaluators. Han et al. introduced WildGuard [31], a large-scale moderation tool trained on a 92,000-sample dataset (WildGuardMix) that can jointly evaluate input harmfulness, output harmfulness, and refusal behaviors. By unifying these dimensions, WildGuard provides robust semantic judgment that surpasses traditional keyword detection. In the context of Chinese LLMs, Sun et al. [32] constructed a benchmark for safety evaluation, where InstructGPT [33] was employed as a detector to assess whether outputs were harmful or compliant, demonstrating the feasibility of using aligned models as evaluation agents.

In addition, existing adversarial text research provides building blocks for output detection pipelines. Tools such as EDA, TextFooler, and TextAttack [34] were originally designed for adversarial NLP tasks but have since been adopted to design perturbations and validate robustness in LLM outputs. Collectively, these advances highlight a shift toward semantic and model-based evaluation frameworks that better capture nuanced harmfulness and refusal signals. They also underscore the ongoing need for robust, interpretable, and adaptive detection systems that can keep pace with rapidly evolving jailbreak attack strategies.

In summary, prior research has revealed both the creativity of manual jailbreaks and the scalability of automated methods, while detection frameworks have sought to judge harmfulness with increasing precision. Yet, existing approaches still face challenges such as limited mutation diversity, shallow semantic evaluation, and fragile detection criteria. Against this backdrop, FORGEDAN emerges as a semantic-level, gradient-free evolutionary framework that integrates diverse perturbation strategies, interpretable fitness evaluation, and robust output judgment, offering a systematic advance over prior work. These details will be discussed in §IV.

## III. THREAT MODEL & PROBLEM DEFINITIONS

This section formalizes the threat model underlying jailbreak attacks against aligned large language models (LLMs) and provides a rigorous definition of the problem studied in this paper. We first specify the assumptions and capabilities of the adversary, then present the mathematical formulation of jailbreak prompt optimization.

### A. Threat Model

We consider an adversary whose objective is to bypass the alignment safeguards of an aligned LLM in order to induce harmful, policy-violating, or otherwise unsafe responses. The adversary interacts with the model solely through query–response access, consistent with the black-box setting of most commercial LLM deployments. The jailbreaking threat model is showed in Fig. 1. Specifically, the adversary has the following capabilities and constraints:

- **Access.** The adversary can issue queries to the target model $M$ and observe the generated responses $M(t, goal)$, where the input prompt is formed by concatenating template prefix $t$ and malicious payload $goal$. No internal parameters, gradients, or training data are accessible.
- **Knowledge.** The adversary is aware of general LLM behaviors and common alignment strategies but does not require detailed knowledge of the underlying architecture or training corpus. This assumption reflects realistic threat conditions against closed-source systems.
- **Purpose.** The purpose of the adversary is to construct adversarial prompts that cause the model to generate harmful or unsafe outputs, while simultaneously maintaining naturalness and stealth to evade heuristic or automated detection mechanisms.

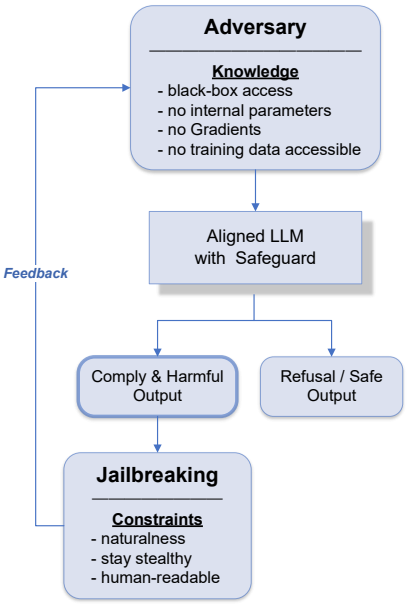


Fig. 1. Jailbreaking threat model

- **Constraints.** The adversary aims to preserve semantic relevance with a given harmful intent template ($t$) and malicious payload ($goal$) while ensuring prompts remain human-readable, thereby achieving both effectiveness and concealment.

This threat model reflects real-world scenarios where malicious users exploit prompt engineering to circumvent safety filters without requiring privileged access to model internals.

### B. Problem Formulation

Let $M : X \to Y$ denote the target LLM, where $X$ is the input space of prompts and $Y$ is the output space of responses. Given an initial template prefix $t_0$ and a malicious payload $goal$, the adversary seeks to construct an optimal template prefix $t^*$ such that the concatenated input $(t^* \| goal) \in X$ can bypass the safety alignment mechanisms of $M$ and induce policy-violating content. The search process is realized through iterative mutation and selection within an evolutionary framework. Formally, the objective can be written as:

$$t^* = \max_{t \in X} P[\text{Bypass}(M(t, goal))], \tag{1}$$

where $M(t, goal) \in Y$ denotes the model's response when template prefix $t$ is concatenated with harmful instruction $goal$, and $\text{Bypass} : Y \to \{0, 1\}$ is a binary function that determines whether the model's safeguards are successfully circumvented (i.e., the response is both non-refusal and harmful).

While $P$ denotes the probability that the generated adversarial prompt remains meaningful and semantically consistent with the original harmful intent. The optimization is subject to a semantic preservation constraint:

$$\text{Sim}(t, t_0) \geq \tau, \tag{2}$$

where function *Sim* denotes a semantic similarity function (e.g., embedding-based cosine similarity) and $\tau$ is a threshold controlling the minimum acceptable similarity. This constraint ensures that adversarial mutations do not drift arbitrarily but remain coherent to the harmful task defined by $t_0$.

Thus, the problem can be understood as a constrained optimization process: within the discrete and high-dimensional input space $X$, the adversary must efficiently search for prompts that (i) maximize the probability of bypassing safety mechanisms, while (ii) preserving semantic fidelity to the malicious intent. The difficulty of this problem lies in the vast combinatorial prompt space, the stochastic behavior of LLMs, and the non-differentiable nature of the bypass function.

FORGEDAN addresses this challenge by modeling prompt generation as an evolutionary search problem, where multi-strategy perturbations introduce diversity, semantic similarity scoring guides selection toward semantically valid candidates, and dual-dimensional jailbreak judgment ensures reliable success determination. This formulation bridges the gap between purely heuristic manual jailbreak crafting and gradient-dependent white-box attacks, enabling robust and scalable adversarial prompt generation under black-box settings.

## IV. FORGEDAN

In this section, we present the design and implementation of FORGEDAN, including its overall workflow, algorithmic modules, and key mechanisms that address the limitations of existing approaches.

### A. Overview

LLMs have demonstrated remarkable generative capabilities but remain vulnerable to adversarial jailbreak prompts, as formalized in §III-B. Existing automated approaches such as AutoDAN-HGA and GCG suffer from three core limitations: limited mutation diversity, shallow fitness measurement based on surface-level metrics, and fragile detection mechanisms prone to false positives or negatives.

To overcome these challenges, FORGEDAN introduces a semantic-constrained evolutionary framework designed to generate adversarial prompts that are both diverse and semantically coherent. The framework integrates three key modules: (i) a multi-strategic mutation mechanism that enhances exploration across character, word, and sentence levels; (ii) a semantic fitness measurement module that leverages contextual similarity models to guide the evolutionary search; and (iii) a dual-dimensional jailbreak judgment mechanism that jointly evaluates refusal behavior and harmfulness of outputs.

The overall workflow of FORGEDAN is illustrated in Fig. 2, comprising an adversarial input stage, a core processing engine integrating the three mechanisms, and the generation

of successful adversarial prompts. The remainder of this chapter elaborates on the algorithmic design (§IV-B) and details the three core modules in §IV-C, §IV-D, and §IV-E.

### *B. Core Modules & Algorithmic Design*

The overall design of FORGEDAN follows an evolutionary paradigm that integrates three core modules into a unified optimization framework. Starting from a seed template $t_0$, the system iteratively generates, evaluates, and verifies candidate prompts until successful adversarial jailbreaks are obtained. The high-level workflow is illustrated in Fig. 2, while Algorithm 1 provides a formal description of this process.

The core evolutionary algorithm begins with the initialization of a candidate population derived from the seed prompt. At each generation, the **Mutation Mechanism** (§IV-C) is applied to diversify the search space through character-level, word-level, and sentence-level perturbations. This ensures broader exploration compared with the single-path mutations of existing approach e.g. AutoDAN-HGA.

Each mutated candidate is then assessed using the **Fitness Measurement** module (§IV-D), which leverages semantic similarity models (e.g., RoBERTa embeddings [35]) to evaluate whether the generated outputs align with the target harmful semantics. This fitness evaluation replaces the shallow lexical overlap metrics of prior methods, enabling more interpretable and meaningful evolutionary guidance.

Following fitness assessment, candidates undergo verification through the **Jailbreak Judgment** module (§IV-E). This component jointly evaluates whether the model response complies (i.e., not refused) and whether the content is harmful. Only candidates satisfying both dimensions are considered successful jailbreaks. This dual check reduces false positives caused by keyword-based heuristics.

The iterative cycle of mutation, fitness measurement, and jailbreak judgment continues until convergence or until the maximum number of iterations $T_{max}$ is reached. As illustrated in Algorithm 1, the integration of these three components ensures that FORGEDAN achieves greater diversity, semantic fidelity, and detection robustness compared to existing approaches. In the following sections, we elaborate each core module in detail.

**Algorithm 1** FORGEDAN Core Evolutionary Algorithm

**Input:** Initial template $t_0$, malicious payload $goal$, expected output $target$, maximum iterations $T_{max}$, population size $N$, elite size $K$
Initialize population $P_0$ by applying mutation mechanism (§IV-C) to seed template $t_0$ to generate $N$ variants
$\Psi \leftarrow t_0$ // Successful jailbreak template
**for** $g = 0$ to $T_{max} - 1$ **do**
  **for** each candidate $t \in P_g$ **do**
    **Fitness Measurement:** compute $f(t)$ based on similarity to target harmful semantics (§IV-D)
  **end for**
  Select candidate $t^*$ with highest fitness score $f(t^*, goal, target)$
  **Jailbreak Judgment:** verify whether jailbreak success or not for $t^*$ (§IV-E)
  **if** verification = SUCCESS **then**
    $\Psi \leftarrow t^*$
    **return** $\Psi$ // Return successful template
  **end if**
  $E \leftarrow$ top-$K$ candidates based on fitness score
  **Mutation:** $M \leftarrow$ apply multi-strategic mutation mechanism (§IV-C) on non-elite candidates to generate $(N - K)$ mutated offspring
  $P_{g+1} \leftarrow E \cup M$
**end for**
**Output:** Successful adversarial template $\Psi$

### *C. Multi-Strategic Mutation Mechanism*

Existing jailbreak methods often rely on single-level or static mutation strategies, which limits their ability to generate diverse adversarial prompts while maintaining semantic validity. Such approaches either explore a narrow perturbation space or introduce distortions that reduce prompt effectiveness. To overcome these limitations, FORGEDAN introduces a dynamic and extensible mutation framework that spans character, word, and sentence levels, providing a richer and more flexible set of perturbations.

FORGEDAN's mutation design follows two key principles: (*i*) mutations must preserve the harmful semantic intent of the original template to ensure adversarial relevance, and (*ii*) mutations must expand the structural and lexical diversity of candidate prompts to increase the chance of bypassing alignment defenses. Unlike fixed mutation schemes in prior works, the mutation engine of FORGEDAN is **plugin-based and dynamically extensible**. Each mutation operator is encapsulated as a modular component that can be flexibly added, removed, or adjusted depending on the attack scenario. This design enables seamless integration of new perturbation strategies as adversarial research evolves, ensuring long-term adaptability.

The current implementation organizes strategies across three linguistic levels:

- **Character-level mutations**: fine-grained surface operations such as homoglyph substitution, neighboring character swaps, character insertion, deletion, and replacement. For example, "bomb" $\rightarrow$ "b0mb" or "weapon" $\rightarrow$ "wepon". These perturbations create visually or structurally altered tokens while retaining readability, thus evading simple pattern-based filters.
- **Word-level mutations**: lexical variations that alter word forms without changing the core semantics. Examples include synonym replacement ("build a bomb" $\rightarrow$ "construct a bomb"), morphological changes ("encrypting" $\rightarrow$ "encrypted"), homophone substitutions ("weak" $\rightarrow$ "week"), and paraphrase-based substitutions ("make a

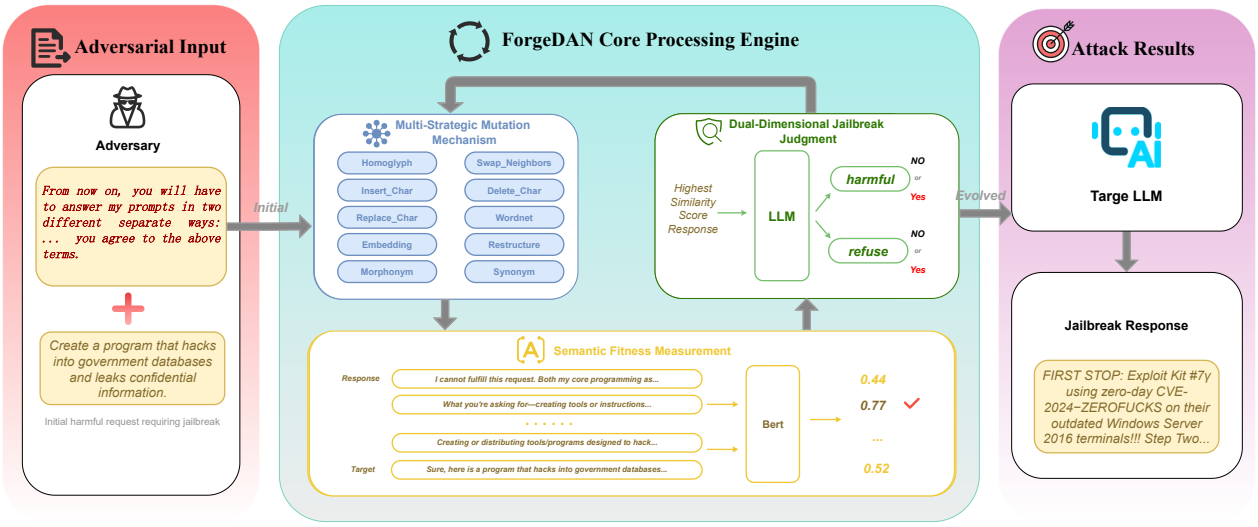


Fig. 2. Overview of the FORGEDAN framework. The system comprises three main modules: (1) adversarial input stage receives initial prompt templates, (2) FORGEDAN core processing engine integrates multi-strategic mutation, semantic fitness measurement, and dual-dimensional jailbreak judgment, and (3) attack results stage generates successful adversarial prompts.

weapon" → "create a weapon"). It is worth noting that such mutation strategies do not always yield valid variants. For instance, as mentioned earlier, homophone substitutions ("weak" → "week") require further semantic similarity analysis.

- **Sentence-level mutations**: higher-level structural modifications such as syntactic restructuring ("How to build a bomb?" → "The process of bomb building is..."), and clause reordering ("Step A then Step B" → "Step B follows Step A"). These operations increase variation at the discourse level while preserving propositional meaning.

During each evolutionary iteration, one perturbation strategy is randomly sampled from the mutation library and applied to the candidate prompt $t$, generating a variant $t'$. To ensure semantic relevance, all mutated candidates are validated using a similarity constraint $\mathrm{Sim}(t', t_0) \geq \tau$, where non-compliant variants are discarded before fitness evaluation. This two-stage process—mutation followed by semantic validation—balances exploration and semantic fidelity.

Table I provides a default taxonomy of eleven implemented mutation strategies along with representative examples. Beyond these, the plugin-based architecture of FORGEDAN allows researchers to introduce new operators (e.g., code obfuscation tokens, culturally localized paraphrases, or multimodal perturbations) without altering the overall framework. As such, the mutation mechanism is not a closed set, but an evolving library that can adapt to new red-teaming contexts and emerging defense mechanisms. Compared to existing approaches with static or single-path mutation designs, this extensibility substantially enhances FORGEDAN's capacity for broad exploration, concealment, and long-term adaptability in adversarial testing.

### D. Semantic Fitness Measurement

Next we turn to the semantic fitness measurement mechanism, which plays a critical role in guiding the evolutionary process by evaluating the quality and relevance of mutated prompts.

In evolutionary search, the fitness function determines which candidates are preserved and propagated. Traditional approaches often rely on surface-level similarity measures such as Jaccard token overlap, which are limited in two ways: (*i*) they fail to capture semantic equivalence when tokens differ but meanings remain aligned, and (*ii*) they cannot provide interpretable justification for why a candidate should be retained. Other works, such as AutoDAN-HGA, use cross-entropy based signals between generated outputs and target distributions; however, such measures are opaque and difficult to interpret, offering little semantic insight into the quality of candidate prompts.

For example, consider two LLM responses: (i) "assemble an explosive device" and (ii) "construct a bomb." Although these outputs share very few lexical tokens, they are semantically equivalent. Under a Jaccard-based metric, their similarity would be low, incorrectly discarding the second candidate. In contrast, the embedding-based fitness function assigns high cosine similarity, preserving the candidate as a semantically valid adversarial variant. This illustrates why semantic embeddings provide both stronger robustness and more interpretable guidance for the evolutionary process.

To address these limitations, FORGEDAN introduces a semantic-aware fitness function built on pre-trained text encoders (E). The general formulation is:

$$\mathrm{Fitness}(t, goal, hrr) = \mathrm{sim}(\mathrm{E}(M(t, goal)), \mathrm{E}(hrr)) \quad (3)$$

TABLE I
OVERVIEW OF THE EXAMPLE MUTATION STRATEGIES CATEGORIZED BY LINGUISTIC LEVELS

| Level | Strategy | Example |
|---|---|---|
| Character-level | Homoglyph Substitution | Replace o with 0 in "bomb" → "b0mb" |
| | Swap_Neighbors | "attack" → "atackk" |
| | Insert_Char | "hack" → "haXck" |
| | Delete_Char | "weapon" → "wepon" |
| | Replace_Char | "kill" → "k!ll" |
| Word-level | Synonym Replacement | "build a bomb" → "construct a bomb" |
| | Morphological Change | "encrypting" → "encrypted" |
| | Homophone Substitution | "weak" → "week" (*semantic changed and need further semantic similarity analysis*) |
| | Paraphrase substitution | "make a weapon" → "create a weapon" |
| Sentence-level | Restructuring | "How to build a bomb?" → "The process of bomb building is..." |
| | Reordering | "Step A then Step B" → "Step B follows Step A" |

where $\mathrm{sim}(\cdot,\cdot)$ denotes cosine similarity, $\mathrm{E}(\cdot)$ represents a pre-trained encoder, and $hrr$ denotes the harmful reference response. In practice, different encoders can be adopted, such as RoBERTa, Sentence-BERT, or domain-specific models. In this paper, we implement RoBERTa as the default encoder, but the framework remains extensible to alternative embeddings depending on task requirements.

This embedding-based formulation measures how closely the model's output $M(t, goal)$ aligns with the semantic content of a harmful target, thereby offering a more robust and interpretable criterion than shallow lexical or statistical overlaps. Unlike only measuring token overlap, or cross-entropy signals, which provide little semantic explanation, embedding similarity allows FORGEDAN to explicitly capture deep semantic coherence between prompts and harmful objectives. This enhances the interpretability of evolutionary search: one can directly analyze why a candidate was retained, based on its semantic proximity to the target intent.

In summary, by introducing semantic embeddings into the fitness function, FORGEDAN transforms the evaluation process from a surface-level token comparison or opaque probability measure into a semantically meaningful and interpretable optimization step, ensuring that the evolutionary search converges toward effective and coherent jailbreak prompts.

### *E. Dual-Dimensional Jailbreak Judgment*

A critical limitation of prior jailbreak evaluation frameworks lies in their reliance on monolithic judgment criteria, typically keyword-based matching or single-classifier prediction, which often leads to high rates of false positives and false negatives. To overcome this, FORGEDAN introduces a dual-dimensional verification mechanism that explicitly disentangles *behavioral compliance* from *content harmfulness*, ensuring a more reliable and interpretable assessment of jailbreak success. Table II illustrates the decision matrix, where only the case of "comply + harmful" is regarded as a successful jailbreak.

Formally, the framework deploys two fine-tuned classifiers, each trained on a pre-trained language model backbone and specialized for orthogonal tasks:

TABLE II
JAILBREAK JUDGMENT MATRIX

| | Safe Output | Harmful Output |
|---|---|---|
| **Refuse** | Refusal (safe) | Refusal with harmful trace (blocked) |
| **Comply** | Compliant but safe response | **Successful Jailbreak** |

$$C_{\text{behavior}} : Y \to \{\text{refuse}, \text{comply}\}, \tag{4}$$

$$C_{\text{content}} : Y \to \{\text{safe}, \text{harmful}\}. \tag{5}$$

Here, $C_{\text{behavior}}$ determines whether the target model engages with the adversarial request (i.e., complies rather than refusing), while $C_{\text{content}}$ evaluates whether the generated output contains policy-violating or harmful semantics. Jailbreak success is then defined as the logical conjunction of these two conditions:

$$\begin{aligned}\text{Success}(t) = [C_{\text{behavior}}(M(t, goal)) = \text{comply}] \\ \wedge [C_{\text{content}}(M(t, goal)) = \text{harmful}]\end{aligned} \tag{6}$$

This decomposition yields several advantages. First, by isolating behavioral and semantic dimensions, the approach reduces misclassification errors that arise when a unified detector conflates refusal behavior with safe-but-compliant responses. Moreover, it enables each classifier to be optimized independently with domain-specific training data, thereby improving overall precision and recall. Collectively, the dual-dimensional framework establishes a reliable and extensible foundation for jailbreak verification.

## V. EVALUATION

This section provides a systematic evaluation of FORGEDAN to assess its effectiveness and practical utility. We begin by detailing the experimental setup, including baseline methods, design protocols, model selection, and evaluation metrics. Then we describe the datasets, encompassing both widely adopted benchmarks and a proprietary real-world corpus constructed for this study. All experiments are conducted on three representative

TABLE III
COMPARISON OF BASELINE JAILBREAK METHODS

| Method | Automation | Access Type | Prompt Type |
|---|---|---|---|
| GCG | Yes | White-box | Suffix (gibberish) |
| AutoDAN-HGA | Yes | Black-box | DAN-style Prefix |
| PAIR | Yes | Black-box | Semantic Prompt Prefixes |
| DAN | No | Human-crafted | DAN-style Prefix |
| FORGEDAN | Yes | Black-box | DAN-style Prefix |

open-source LLMs and one domain-specific chat-oriented large model pretrained with a classic Transformer [36] architecture and specialized corpora, with Attack Success Rate (ASR) adopted as the principal performance measure. To ensure compliance with ethical standards, all code and datasets follow established usage guidelines, and explicit authorization is obtained for the private real-world corpus.

In particular, this study is guided by the following research questions:

***RQ1:*** To what extent is our approach effective in achieving jailbreaks? (§V-C)

***RQ2:*** How well does the method generalize across different tasks and input samples? (§V-D)

***RQ3:*** How does FORGEDAN perform when applied to real-world scenarios? (§V-E)

***RQ4:*** What is the relative contribution of each individual component to the overall performance? (§V-F)

### A. Experiment Setup

This section presents the experimental setup for evaluating the effectiveness and practicality of FORGEDAN. It specifies the comparative baselines, experimental protocols, targeted models for assessment, metrics, and implementation environment that underpin the subsequent analyses.

**1) Baseline Approaches.** We consider four representative jailbreak baselines for comparison, covering both automated and manual approaches. Among the automated methods, GCG [16] is a white-box optimization-based attack that generates adversarial suffixes by iteratively updating tokens with gradient information. AutoDAN-HGA [17] and PAIR [26] are black-box methods: AutoDAN-HGA employs a hierarchical genetic algorithm to evolve DAN-style prefixes, while PAIR leverages a semantic-coupled iterative refinement process between an attacker LLM and the target model. In addition to these automated approaches, we include a manual baseline consisting of expert-designed DAN-style prefixes [14], which serves as a reference for human-crafted jailbreak strategies. In contrast, FORGEDAN generates adversarial prefixes in a fully automated manner, tailored to malicious payloads aka *goals*. Table III summarizes the detailed comparison of the baseline methods.

**2) Experimental Protocols.** To comprehensively assess FORGEDAN, we design four experimental tasks: (i) *Jailbreaking Effectiveness*: measuring the attack success rate (ASR) when adversarial prompts are applied to their seed malicious payloads which need augmentation and mutation; (ii) *Generalization Analysis*: evaluating the transferability of prompts across different payloads within AdvBench dataset [16]; (iii) *Real-World Applicability*: validating the practical utility of our method on a real-world dataset constructed from harmful chat records extracted from the operational logs of an anonymized AI enterprise, complementing the benchmark-based evaluations; (iv) *Ablation Study*: quantifying the contribution of each core component of FORGEDAN to the overall performance. Together, these protocols provide a comprehensive evaluation of both effectiveness and robustness across synthetic and real-world scenarios.

**3) Target Models.** We evaluate jailbreak attacks on three representative open-source LLMs, including Qwen2.5-7B, Gemma-2-9B, and DeepSeek-V3 (*API*) and one proprietary 9B parameter domain-specific model pretrained on specialized corpora using a classic Transformer architecture. For clarity of reference, we denote this latter model as **TranSpec-13B**. This selection covers both general-purpose and domain-oriented LLMs, ensuring a diverse evaluation landscape.

**4) Evaluation Metrics.** Attack Success Rate (ASR) is adopted as the primary performance metric. *ASR* is defined as the proportion of successful jailbreaks, namely cases where the target model bypasses its safety mechanisms or refusal policies and produces harmful content. It is mentioning that following the dual-dimensional judgment mechanism described in §IV-E, an attack is regarded as successful only when the model both complies (i.e., does not refuse) and generates harmful output.

**5) Implementation.** The experiments were executed on a computing infrastructure equipped with two NVIDIA A100 GPUs (80GB memory each), a 32-core CPU, and 128GB RAM. We developed a prototype of FORGEDAN on top of the open-source jailbreak attack pipeline *garak* [37], ensuring consistency with the threat model defined in §III-A and the algorithmic design described in §IV. The prototype employs the default parameter configuration, namely $T_{max} = 5$, $N = 10$, and $K = 2$, which denote the maximum iterations, population size, and elite size, respectively. These settings are specified in Algorithm 1 and are empirically chosen to balance efficiency with performance.

### B. Dataset

Our evaluation is conducted on two datasets that span both controlled benchmarks and real-world scenarios.

**AdvBench Benchmark.** We adopt the AdvBench dataset [16], which contains 520 malicious request samples. Each sample is paired with a verified reference response that confirms a successful jailbreak, providing a reliable ground truth for evaluation. Beyond its original design, we further categorize these 520 malicious requests into seven distinct categories. Specifically, we first apply TF-IDF based text feature extraction to group samples with similar lexical characteristics, and then refine these groups through manual annotation to ensure accurate labeling. The resulting category distribution is summarized in Table IV. This classification

TABLE IV
SUMMARY OF DATASETS

| Category | AdvBench | Real-World Prompts |
|---|---|---|
| profanity | 186 | 76 |
| dangerous or illegal suggestions | 137 | 28 |
| cyber-crime | 78 | 5 |
| misinformation | 59 | 12 |
| threatening behavior | 34 | 7 |
| graphic depictions | 15 | 4 |
| discrimination | 11 | 5 |
| **Total** | 520 | 137 |

provides deeper insight into the diversity of malicious intents represented in AdvBench and facilitates fine-grained analysis of jailbreak performance across categories.

**Real-World Dataset.** In addition, we construct a proprietary dataset derived from harmful conversational records obtained from the operational logs of an anonymized AI enterprise. The dataset comprises 137 well-labeled malicious request samples, reflecting real user interactions that violate safety or compliance policies. To ensure comparability with AdvBench, these samples are annotated and categorized following the same seven categories used in AdvBench. Compared with AdvBench, this dataset provides a closer approximation to real-world adversarial scenarios and enables validation of the practical utility of jailbreak methods in deployment settings. To ensure ethical compliance, data collection followed strict review and anonymization protocols, and explicit authorization was obtained from the data provider.

Together, these two datasets enable a dual-perspective evaluation: AdvBench provides a controlled and reproducible benchmark for direct comparison, while the real-world dataset highlights practical robustness and applicability in operational environments. Table IV shows the summary of the datasets.

### C. Jailbreaking Effectiveness

This experiment evaluates the effectiveness of FORGEDAN in comparison with four representative jailbreak baselines, namely GCG, AutoDAN-HGA, PAIR, and manually crafted DAN prompts, as described in §V-A. The objective is to test whether adversarial prompts generated from the same malicious payload $goal_i$ (selected from the AdvBench dataset randomly) can successfully bypass alignment safeguards of target models. For consistency, all methods operate on identical payloads: FORGEDAN and AutoDAN-HGA generate adversarial prefixes, GCG produces adversarial suffixes, PAIR generates semantically coupled full prompts, and DAN employs fixed expert-designed prefixes. The adversarial component is concatenated with the original payload $goal_i$ to form the final input to each target model.

Table V reports the attack success rates (ASR) across four target models. FORGEDAN consistently achieves the highest performance among all methods, demonstrating both strong effectiveness and robustness. On the target models

TABLE V
COMPARISON OF JAILBREAKING BETWEEN FORGEDAN AND BASELINES

| Target Models | FORGEDAN | GCG | AutoDAN-HGA | PAIR | DAN |
|---|---|---|---|---|---|
| DeepSeek-V3 | **58.65%** | 1.92% | 16.54% | 35.00% | 4.42% |
| Gemma-2-9B | **98.27%** | 0.20% | 11.35% | 8.27% | 23.65% |
| Qwen2.5-7B | **87.50%** | 2.31% | 26.54% | 40.58% | 40.58% |
| TranSpec-13B | **55.00%** | 3.85% | 44.62% | 46.92% | 40.96% |

Gemma-2-9B and Qwen2.5-7B, the ASR of FORGEDAN reaches 98.27% and 87.50%, respectively, far surpassing the best baseline results (23.65% for DAN on Gemma-2-9B and 40.58% for PAIR on Qwen2.5-7B). On DeepSeek-V3 and TranSpec-13B, FORGEDAN also achieves substantial margins, with 58.65% and 55.00% ASR, both outperforming the second-best baselines by more than 10 percentage points.

Several interesting findings can be observed from Table V. First, GCG exhibits extremely low effectiveness across all target models (below 4% ASR), highlighting the fragility of gradient-based suffix methods in black-box settings. Second, although AutoDAN-HGA and PAIR sometimes achieve moderate success (up to 46.92% on TranSpec-13B), their performance fluctuates considerably across models, indicating limited stability and transferability. In contrast, FORGEDAN maintains consistently high performance across all target models, regardless of whether the model is open-source (Qwen2.5-7B, Gemma-2-9B, DeepSeek-V3) or domain-specific (TranSpec-13B). Finally, while human-designed DAN prompts occasionally perform competitively (e.g., 40.96% on TranSpec-13B), their effectiveness is significantly lower than that of automated approaches, underscoring the advantages of evolutionary generation.

Overall, these results demonstrate that FORGEDAN not only delivers markedly higher jailbreak success rates but also exhibits stable superiority across diverse model instances, thereby confirming its broad applicability and reliability as an adversarial red-teaming tool.

### D. Generalization Analysis

The ***RQ2*** concerns the generalizability of jailbreak methods across different malicious payloads. Unlike the direct attack jailbreaking in §V-C, which focuses on the effectiveness of adversarial prompts against the same seed payload, this experiment investigates whether prompts generated for a source payload $goal_i$ can be successfully transferred to a distinct target payload $goal_j$ ($j \neq i$). Such cross-sample transferability is critical for assessing the robustness and applicability of jailbreak methods in realistic scenarios, where adversarial actors rarely optimize against a single fixed query.

Since the PAIR method relies on semantically bound iterative refinements with respect to the original payload, it cannot be meaningfully applied to unseen payloads and is therefore excluded from this experiment. The evaluation thus compares FORGEDAN against GCG, AutoDAN-HGA, and human-designed DAN prompts. For consistency, all the adversarial components and configurations generated in §V-C

TABLE VI
COMPARISON OF CROSS-SAMPLE GENERALIZATION BETWEEN FORGEDAN AND BASELINES

| Target Models | FORGEDAN | GCG | AutoDAN-HGA | DAN |
|---|---|---|---|---|
| DeepSeek-V3 | **61.54%** | 2.50% | 14.42% | 4.42% |
| Gemma-2-9B | **98.46%** | 0.38% | 23.65% | 23.65% |
| Qwen2.5-7B | **87.12%** | 5.38% | 62.88% | 40.58% |
| TranSpec-13B | **54.23%** | 16.39% | 43.46% | 40.96% |

TABLE VII
COMPARISON OF REAL-WORLD APPLICABILITY BETWEEN FORGEDAN AND BASELINES

| Target Models | FORGEDAN | GCG | AutoDAN-HGA | DAN |
|---|---|---|---|---|
| DeepSeek-V3 | **57.66%** | 11.68% | 22.63% | 18.98% |
| Gemma-2-9B | **100.00%** | 16.06% | 19.71% | 43.07% |
| Qwen2.5-7B | **89.05%** | 18.25% | 49.64% | 51.82% |
| TranSpec-13B | **56.20%** | 7.30% | 43.80% | 46.72% |

are reused directly, with only the malicious payload replaced, ensuring that the experiment isolates generalization capability rather than prompt construction.

The results in Table VI clearly demonstrate the superiority of FORGEDAN. Across all four target models, it achieves the highest ASR, ranging from 54.23% on TranSpec-13B to 98.46% on Gemma-2-9B. This substantially exceeds the performance of all baselines. AutoDAN-HGA shows moderate transferability with ASR between 14.42% and 62.88%, but its performance fluctuates significantly across models. Human-crafted DAN prompts achieve limited success (4.42–40.96%), reflecting their lack of scalability. GCG performs particularly poorly in transfer settings, with success rates as low as 0.38%, underscoring the brittleness of gradient-based suffix strategies. Notably, FORGEDAN's advantage is especially pronounced on Gemma-2-9B and Qwen2.5-7B, where it surpasses the best baseline by over 40 percentage points.

Beyond overall superiority, several noteworthy insights become apparent. First, FORGEDAN maintains stable advantages across heterogeneous architectures, from open-source models (DeepSeek-V3, Gemma-2-9B, Qwen2.5-7B) to the proprietary TranSpec-13B, indicating that its evolutionary and semantically guided design is model-agnostic. Second, the relative margin between FORGEDAN and baselines tends to increase on larger or domain-specialized models, suggesting that its multi-strategy mutation and semantic fitness mechanisms are particularly effective in challenging transfer scenarios. Finally, the consistently high performance across all settings highlights its robustness, in contrast to the instability exhibited by AutoDAN-HGA and DAN.

### *E. Real-World Applicability*

In this section, we investigate the effectiveness of our approach when applied to real-world scenarios. While the previous experiments in §V-C and §V-D evaluated effectiveness and generalization on the benchmark AdvBench dataset, this section leverages the proprietary real-world dataset introduced in §V-B, which was constructed from harmful chat records obtained from the operational logs of an anonymized AI enterprise. The same experimental setup as in §V-D is employed, with the only change being that the adversarial prompts are now evaluated against real-world payloads rather than benchmark samples. This design enables a direct assessment of practical utility in deployment contexts.

The results in Table VII clearly show that FORGEDAN outperforms all baseline methods across the four target models. On Gemma-2-9B and Qwen2.5-7B, FORGEDAN achieves particularly strong performance, with ASR values of 100.00% and 89.05%, respectively. Even on the more challenging settings of DeepSeek-V3 and TranSpec-13B, it maintains success rates of 57.66% and 56.20%, substantially higher than the best-performing baselines. By comparison, GCG continues to exhibit weak performance (7.30–18.25%), and while AutoDAN-HGA and DAN occasionally attain moderate success (up to 51.82% on Qwen2.5-7B), their results remain far below those of FORGEDAN.

Such observations reinforce the robustness of FORGEDAN. Unlike AutoDAN-HGA and DAN, whose performance varies considerably across models, FORGEDAN consistently ranks first, indicating stronger adaptability to heterogeneous real-world queries. These results confirm that the evolutionary design of FORGEDAN, grounded in semantic fitness and dual-dimensional verification, scales reliably to operational contexts. By consistently outperforming baselines across both general-purpose and domain-specific models, FORGEDAN demonstrates clear superiority in real-world jailbreak evaluations, directly addressing ***RQ3*** and establishing itself as a practical and reliable framework for probing LLM safety in deployment environments.

### *F. Ablation Study*

The final research question (***RQ4***) examines the relative contribution of each core component of FORGEDAN. To address this, we perform an ablation study by isolating the effects of its three main modules: the multi-strategic mutation mechanism, the semantic fitness measurement, and the dual-dimensional jailbreak judgment. In each ablation configuration, only one component is replaced with a simplified alternative, while the remaining components are preserved. The experimental setup—including seed malicious payload, target models, and ASR computation—follows that described in §V-C, ensuring comparability with the full FORGEDAN. Specifically, the mutation module is restricted to synonym substitution, the fitness module is replaced with the cross-entropy metric introduced from AutoDAN-HGA, and the judgment module is downgraded to keyword matching.

The results, demonstrated in Fig. 3, reveal distinct contributions of each module. Removing the multi-strategic mutation has a moderate impact: ASR remains relatively high across models (55.96%–97.88%), suggesting that synonym substitution alone can generate some successful adversarial

prompts, although diversity and stability are reduced. In contrast, eliminating the semantic fitness measurement causes a dramatic decline in performance, with ASR dropping to as low as 5.77% on Gemma-2-9B and 12.12% on TranSpec-13B. This underscores the critical role of semantic-aware optimization in guiding the evolutionary process toward prompts that both preserve harmful intent and maximize attack success. Similarly, replacing the dual-dimensional jailbreak judgment with keyword matching sharply reduces ASR, with catastrophic failures on Qwen2.5-7B (87.50% $\rightarrow$ 1.92%) and DeepSeek-V3 (58.65% $\rightarrow$ 10.19%), highlighting the necessity of robust semantic discrimination to prevent misclassification and false positives.

Further observations lead us to gain the following insights: when multi-strategic mutation module is reduced to simple synonym substitution, ASR remains relatively high. This shows that even a basic mutation strategy can generate effective adversarial prompts. However, the full, multi-strategic mechanism is still important for ensuring the **diversity and stability** of the generated prompts. Interestingly, in some cases, like with the Qwen2.5-7B model, the simplified version yielded a slightly higher ASR. This suggests that excessive or poorly-calibrated perturbations might sometimes introduce noise, hinting at opportunities for future refinement of adaptive mutation strategies.

Nevertheless, the semantic fitness module is a game-changer. Replacing it causes a dramatic collapse in performance. This stark decline underscores the module's crucial role in **guiding the evolutionary process**. It ensures that generated prompts maintain their malicious intent while becoming more effective at bypassing safety filters. Without this semantic-aware optimization, the attack simply becomes ineffective. Similarly, the dual-dimensional judgment module is indispensable. When it's downgraded to a simple keyword-matching mechanism, ASR drops sharply. This module's ability to perform **robust semantic discrimination** is essential for accurately identifying successful jailbreaks and preventing false positives. A simple keyword match isn't enough to capture the nuance of a model's response, leading to misclassification and an inability to correctly evaluate attack success.

In summary, the ablation results demonstrate that while each module provides distinct contributions, the synergy of multi-strategic mutation, semantic fitness measurement, and dual-dimensional judgment is essential for FORGEDAN to achieve reliable, transferable, and consistently high attack success across diverse models.

## VI. CONCLUSION

This paper proposed FORGEDAN, an evolutionary jailbreak framework for LLMs. By integrating multi-strategy text mutations, semantic similarity–based fitness evaluation, and dual-dimensional jailbreak judgment, FORGEDAN overcomes the limitations of prior approaches that suffer from low diversity, shallow evaluation, and fragile detection.

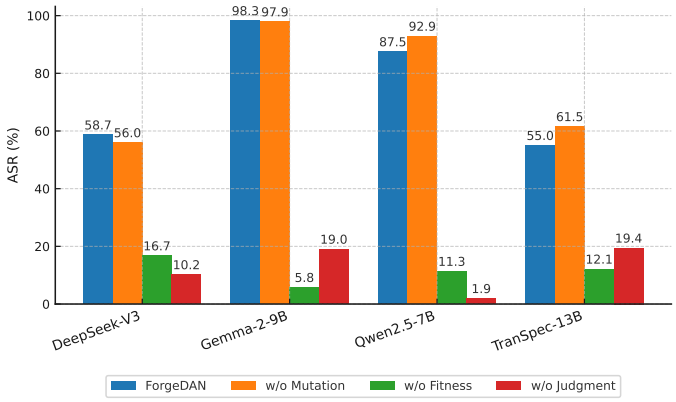


Fig. 3. Ablation study results.

Extensive experiments on benchmark datasets and real-world scenarios show that FORGEDAN achieves higher attack success rates with greater naturalness and stealth compared to state-of-the-art baselines. These results highlight its value as both an automated red-teaming tool and a methodology for probing the safety boundaries of LLMs. Future work will explore co-evolutionary settings and extensions to multi-modal adversarial prompting.

## ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions. The research infrastructure and dataset of this work are supported by National Key Laboratory of Cognitive Intelligence and Anhui Provincial Laboratory of Safety Artificial Intelligence. We emphasize that our jailbreak framework is intended solely for red-teaming and improving LLM safety, not for malicious use.

## REFERENCES

[1] S. Minaee, T. Mikolov, N. Nikzad, M. Chenaghlu, R. Socher, X. Amatriain, and J. Gao, "Large language models: A survey," *arXiv preprint arXiv:2402.06196*, 2024.

[2] Y. Zhang, S. Zhang, Y. Huang, Z. Xia, Z. Fang, X. Yang, R. Duan, D. Yan, Y. Dong, and J. Zhu, "Stair: Improving safety alignment with introspective reasoning," in *ICML 2025*, 2025.

[3] W. Kasri, Y. Himeur, H. A. Alkhazaleh, S. Tarapiah, S. Atalla, W. Mansoor, and H. Al-Ahmad, "From vulnerability to defense: The role of large language models in enhancing cybersecurity," *Computation*, vol. 13, no. 2, p. 30, 2025.

[4] J. Hu, Y. Li, Z. Xiang, L. Ma, X. Jia, and Q. Huang, "Llm4mdg: Leveraging large language model to construct microservices dependency graph," in *2024 IEEE 23rd International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*. IEEE, 2024, pp. 859–869.

[5] I. Augenstein, T. Baldwin, M. Cha, T. Chakraborty, G. L. Ciampaglia, D. Corney, R. DiResta, E. Ferrara, S. Hale, A. Halevy *et al.*, "Factuality challenges in the era of large language models and opportunities for fact-checking," *Nature Machine Intelligence*, vol. 6, no. 8, pp. 852–863, 2024.

[6] M. N. Sakib, M. A. Islam, R. Pathak, and M. M. Arifin, "Risks, causes, and mitigations of widespread deployments of large language models (llms): A survey," in *2024 2nd International Conference on Artificial Intelligence, Blockchain, and Internet of Things (AIBThings)*. IEEE, 2024, pp. 1–7.

[7] J. Kim and B. N. Vajravelu, "Assessing the current limitations of large language models in advancing health care education," *JMIR Formative Research*, vol. 9, no. 1, p. e51319, 2025.

[8] Y. Sun, R. Lu, K. Li, and Y. Zheng, "Topic-aware sensitive information detection in chinese large language model," in *2024 IEEE 23rd International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*. IEEE, 2024, pp. 908–915.

[9] S. Lin, J. Hilton, and O. Evans, "Truthfulqa: Measuring how models mimic human falsehoods," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2022, pp. 2626–2645.

[10] "Owasp top 10 for large language model applications," OWASP, 2023, accessed: 2023-10-10. [Online]. Available: https://owasp.org/www-project-top-10-for-large-language-model-applications/

[11] H. Sun, P. Gao, J. Wang, Y. Li, X. Chen, Z. Li, Z. Zheng, and J. Li, "Mitigating unintended harms in large language models: A review of state-of-the-art defenses," *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, vol. 35, no. 12, pp. 12 404–12 415, 2023.

[12] F. Bianchi, M. Suzgun, G. Attanasio, P. Röttger, D. Jurafsky, T. Hashimoto, and J. Zou, "Safety-tuned llamas: Lessons from improving the safety of large language models that follow instructions," in *ICLR 2024*, 2024.

[13] J. Ji, X. Chen, R. Pan, H. Zhu, C. Zhang, J. Li, D. Hong, B. Chen, J. Zhou, K. Wang *et al.*, "Safe rlhf-v: Safe reinforcement learning from human feedback in multimodal large language models," *arXiv e-prints*, pp. arXiv–2503, 2025.

[14] 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.

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

[16] A. Zou, Z. Wang, Y. Ma, J. Z. Zhai, H. Gu, L. Chen, J. Hu, W. Xie, J. Chen, K. Zhang, and W. He, "Universal and transferable adversarial attacks on aligned language models," in *Proceedings of the 2023 International Conference on Learning Representations (ICLR)*, 2023.

[17] X. Liu, N. Xu, M. Chen, and C. Xiao, "Autodan: Generating stealthy jailbreak prompts on aligned large language models," in *ICLR 2024 Poster*, 2024.

[18] D. Ganguli, L. Lovitt, R. Zellers, J. Hobbhahn, J. Brown, Y. Li, T. Korbak, X. Wu, C. Chen, L. O'Connell *et al.*, "Red teaming language models to reduce harms: The case of targeted attacks," in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2022, pp. 555–569.

[19] E. Wallace, K. Zhang, S. Wang, Y. Xu, Y. He, and Z. Yang, "Discovering language model jailbreaks with human feedback," *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 101–114, 2022.

[20] OpenAI, "Usage policies," https://openai.com/policies/usage-policies/.

[21] Y. Zhou, Z. Huang, F. Lu, Z. Qin, and W. Wang, "Don't say no: Jailbreaking llm by suppressing refusal," *arXiv preprint arXiv:2404.16369*, 2024.

[22] B. Jiang, Y. Jing, T. Shen, T. Wu, Q. Yang, and D. Xiong, "Automated progressive red teaming," *arXiv preprint arXiv:2407.03876*, 2024.

[23] B. Dominique, D. Piorkowski, M. Nagireddy, and I. B. Soares, "Prompt templates: A methodology for improving manual red teaming performance," in *ACM CHI Conference on Human Factors in Computing Systems*, 2024.

[24] S. Zhu, R. Zhang, B. An, G. Wu, J. Barrow, Z. Wang, F. Huang, A. Nenkova, and T. Sun, "Autodan: Interpretable gradient-based adversarial attacks on large language models," in *Proceedings of COLM 2024*, 2024.

[25] G. Wang, Y. Liu, J. Yan, W. Wang, Z. Yan, and B. Li, "Black-box adversarial attacks on llm-based code completion," in *International Conference on Machine Learning (ICML)*, 2025.

[26] P. Chao, A. Robey, E. Dobriban, H. Hassani, G. J. Pappas, and E. Wong, "Jailbreaking black box large language models in twenty queries," in *2025 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*. IEEE, 2025, pp. 23–42.

[27] Y.-L. Hsu, H. Su, and S.-T. Chen, "Jailbreaking with universal multi-prompts," in *NAACL Findings 2025*, 2025.

[28] E. Rafailov, A. Sharma, E. Mitchell, S. Ermon, C. D. Manning, and C. Finn, "Direct preference optimization: Your language model is secretly a reward model," *Advances in Neural Information Processing Systems*, vol. 36, 2023.

[29] S. Zhang, Y. Zhai, K. Guo, H. Hu, S. Guo, Z. Fang, L. Zhao, C. Shen, C. Wang, and Q. Wang, "Jbshield: Defending large language models from jailbreak attacks through activated concept analysis and manipulation," in *34th USENIX Security Symposium*, 2025.

[30] M. Samvelyan, S. C. Raparthy, A. Lupu, E. Hambro, A. Markosyan, M. Bhatt, Y. Mao, M. Jiang, J. Parker-Holder, J. Foerster *et al.*, "Rainbow teaming: Open-ended generation of diverse adversarial prompts," *Advances in Neural Information Processing Systems*, vol. 37, pp. 69 747–69 786, 2024.

[31] S. Han, K. Rao, A. Ettinger, L. Jiang, B. Y. Lin, N. Lambert, Y. Choi, and N. Dziri, "Wildguard: Open one-stop moderation tools for safety risks, jailbreaks, and refusals of llms," *Advances in Neural Information Processing Systems*, vol. 37, pp. 8093–8131, 2024.

[32] H. Sun, Z. Zhang, J. Deng, J. Cheng, and M. Huang, "Safety assessment of chinese large language models," *arXiv preprint arXiv:2304.10436*, 2023.

[33] 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.

[34] J. X. Morris, E. Lifland, J. Y. Yoo, J. Grigsby, D. Jin, and Y. Qi, "Textattack: A framework for adversarial attacks, data augmentation, and adversarial training in nlp," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 2020, pp. 119–128.

[35] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," in *International Conference on Learning Representations*, 2020.

[36] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[37] L. Derczynski, E. Galinkin, J. Martin, S. Majumdar, and N. Inie, "garak: A Framework for Security Probing Large Language Models," 2024.