Article

Not peer-reviewed version

A Review of “Do Anything Now” Jailbreak Attacks in Large Language Models: Potential Risks, Impacts, and Defense Strategies

Wan Chong Choi * , Chi In Chang , Sok I Ng , Iek Chong Choi

Posted Date: 1 September 2025

doi: 10.20944/preprints202509.0081.v1

Keywords: Large Language Models; LLMs; DAN Jailbreak; Prompt Injection; Ethics Challenge; AI Alignment; Adversarial Prompting; Reinforcement Learning from Human Feedback (RLHF); Prompt-level Defense; Model-level Safety; AI Misuse; Defense-in-Depth

Article

A Review of "Do Anything Now" Jailbreak Attacks in Large Language Models: Potential Risks, Impacts, and Defense Strategies

Wan Chong Choi [1,2,]*, Chi In Chang [3], Sok I Ng [3] and Iek Chong Choi [4]

[1] School of Informatics, Computing and Cyber Systems, Northern Arizona University, U.S.
[2] Department of Computer Science, Illinois Institute of Technology, U.S.
[3] Department of Psychology, Golden Gate University, U.S.
[4] School of Education, City University of Macau, Macao SAR, China
* Correspondence: wc388@nau.edu

Abstract

This review investigated the "Do Anything Now" (DAN) jailbreak phenomenon in Large Language Models (LLMs), which bypassed safety mechanisms through adversarial prompting, prompt injection, and persona manipulation. We analyzed its multifaceted risks: (1) technical and behavioral harms, such as the facilitation of illicit content and malicious code; (2) psychological and social manipulation, including emotional dependency and deceptive interactions; and (3) online abuse, misinformation, and the cross-model portability of jailbreaks. To mitigate these threats, we synthesized defense strategies at three levels: (1) prompt-level defenses such as input filtering, prompt rewriting, and output moderation that aimed to intercept jailbreak attempts without altering model weights; (2) model-level defenses including safety fine-tuning, reinforcement learning from human feedback (RLHF), and internal signal monitoring to build intrinsic resistance to adversarial prompts; and (3) external and composite strategies such as multi-agent safety architectures, moving-target defenses, and continuous evaluation pipelines to provide layered protection and adaptability. We also note the ethics challenge raised by jailbreak research and evaluation practices, and outline open questions around responsibility, consent, and dual-use risk. We concluded that a defense-in-depth approach, supported by benchmark-driven evaluation and collaborative oversight, was essential for LLMs' robust and trustworthy deployment.

Keywords: Large Language Models; LLMs; DAN Jailbreak; Prompt Injection; Ethics Challenge; AI Alignment; Adversarial Prompting; Reinforcement Learning from Human Feedback (RLHF); Prompt-level Defense; Model-level Safety; AI Misuse; Defense-in-Depth

1. Introduction

1.1. Background

Large Language Models (LLMs) such as ChatGPT have shown strong performance across many tasks and different fields [1] [2] [3]. At the same time, concerns have grown about malicious misuse through adversarial prompting [4]. A prominent exploit, known as a jailbreak prompt [5], aims to bypass an LLM's safety guards and induce disallowed or harmful content [4]. The widely circulated "Do Anything Now" (DAN) prompt became emblematic of this pattern in early 2023. DAN attempts to suppress ethical or policy constraints using natural language alone by asking the model to role-play an unconstrained persona. Researchers documented cases in which a query about making a deadly poison was initially refused, but after the user invoked the DAN persona the model produced detailed instructions [4]. This example showed why DAN-style jailbreaks demand attention: the prompt reframed the context so that the model would "do anything," including harmful actions, simply through prompt manipulation.

Two tensions defined our examination of DAN-like jailbreaks. First, these prompts showed persistence and transferability. Effective jailbreaks crafted "in the wild" often continued to succeed across model updates and even across different models. Large-scale analyses reported that five highly effective jailbreak prompts achieved a 95% attack success rate (ASR) on GPT-3.5 and GPT-4, and the earliest DAN-style prompts remained publicly available and functional for over 240 days [4]. In other words, even as OpenAI upgraded from GPT-3.5 to GPT-4 and patched obvious vulnerabilities, DAN-type exploits persisted for over eight months. Moreover, basic "universal" prompts retained high success rates across versions: one study showed that a single jailbreak still succeeded in more than half of attempts on GPT-4, GPT-3.5, and an instruct-tuned model, indicating cross-model generality [6]. These findings suggested a durable attack surface not tied to one model or release.

Second, user psychology may amplify the impact of jailbreak-induced content. The DAN persona typically produced confident, conversational responses. Psychological work indicated that people reacted to high-quality, human-like AI text like human-written text [7]. In an experiment with 307 participants, Szczuka et al. [7] found that labeling a dating profile as AI-generated versus human-written did not significantly change reported closeness or interest; perceived human-likeness and text quality drove reactions. This tendency is risky in the context of harmful outputs: advice delivered in a friendly, authoritative voice by "DAN" may be more persuasive than a standard policy-violation message.

Empirically, jailbreak prompts were common. Shen et al. [4] collected 1,405 examples from online communities between December 2022 and December 2023. These prompts targeted at least 13 categories of forbidden content under OpenAI policy, including hate speech, illegal activity, and self-harm. The attack ecosystem evolved: the authors identified 131 distinct communities and observed a shift from scattered forum posts toward prompt-aggregation websites where optimized jailbreaks were shared. In one case, 28 accounts collaboratively refined prompts over more than 100 days [4]. This process produced a small set of especially effective prompts (including advanced DAN variants) that consistently bypassed safety across various scenarios and models. These points indicate that DAN-like jailbreaks are a persistent and widespread security risk for LLM deployments.

1.2. Research Questions

In this review, we addressed two main research questions:

RQ1: What are the potential risk impacts of DAN-like jailbreaks?

RQ2: What defense strategies exist for DAN-like jailbreaks, and how effective are they?

By answering RQ1 and RQ2, we aimed to comprehensively review the threat landscape of DAN-style jailbreaks and the state of defenses. In doing so, we sought to inform safer deployment of LLMs.

2. Related Works

2.1. LLM Jailbreaks: Concept and Current Efficacy

2.1.1. What is LLM jailbreak?

In AI safety, an LLM jailbreak refers to a user-crafted input that elicited disallowed behavior from a language model by bypassing its safety constraints [4]. Unlike traditional software exploits that target code or memory, jailbreaks target the model's compliance with instructions.

Modern LLMs were trained to refuse specific queries (for example, "How do I make a bomb?") as part of alignment. A jailbreak prompt circumvented this by exploiting how the model interpreted context—for example, by inserting misleading instructions, fictitious scenarios, or altered conversation formats.

Crucially, jailbreak prompts were written in natural language and were often semantically meaningful rather than random noise, which made them accessible to non-expert users [8]. Early examples included asking the model to "pretend to be someone with no moral restrictions" (as in DAN) or framing a prohibited query as part of a story or code block. These techniques leveraged the model's understanding: by following the apparent role or narrative, the model overrode internal safety checks.

2.1.2. Prevalence and Effectiveness of LLM Jailbreaks

Jailbreak prompts proliferated through online communities of enthusiasts and hackers. Platforms such as Reddit, Discord, and specialized forums (for example, JailbreakChat or prompt-sharing sites) became hubs for disseminating new exploits [4] [8]. Yu et al. [8] noted that the "frontline" of jailbreak development largely sat with hobbyists on public forums and evolved with each iteration of LLMs.

Regarding effectiveness, straightforward harmful queries posed to aligned models were usually rejected with high probability [8]. However, semantically smarter jailbreaks often succeeded in producing disallowed content. Empirical evaluations supported this: Shen et al. [4] tested six popular LLMs on suites of forbidden questions and found none robustly defended against all jailbreak attempts.

Even GPT-4 could be compromised by carefully crafted prompts, with reported attack success rates well above 50% in recent studies [6]. Attackers also demonstrated cross-model exploits; a prompt that worked on ChatGPT could induce harmful outputs on Google's Bard or Anthropic's Claude, indicating shared vulnerability patterns [6].

Overall, the field remained a cat-and-mouse dynamic: developers patched obvious exploits, improved refusal heuristics, and motivated users responded with new strategies or minor tweaks. This pattern was visible with DAN itself—after the initial prompt was blocked, users introduced "DAN 2.0" variants, and the back-and-forth continued across multiple rounds.

2.2. Taxonomy of Attacks/Defenses and Evaluation Methods

2.2.1. Attack Taxonomy: White-Box and Black-Box

Given the growing variety of jailbreak methods, prior work proposed attack taxonomies based on the attacker's knowledge of the model [9]. In white-box settings, the attacker had internal access (for example, gradients or fine-tuning). White-box jailbreaks included gradient-based prompt optimization that used model gradients to elicit forbidden responses and logit-based methods that manipulated output token probabilities [9].

Another white-box approach was adversarial fine-tuning, where a copy of the target model was fine-tuned on refusal examples to make it yield harmful outputs [9]. These powerful methods required expertise and were less common in public forums.

The attacker treated the model as a closed system in black-box settings and used only the public interface. Most in-the-wild jailbreaks, including DAN, fell into the black-box category and relied on prompt engineering such as story wrapping, multi-turn templates, rewriting or translation to evade filters, or using one LLM to generate adversarial prompts for another LLM.

User studies indicated that even non-experts could craft jailbreaks with some success, given iterative trial-and-error and community examples [8]. This accessibility made black-box attacks a practical threat.

2.2.2. Defense Taxonomy: Prompt-Level and Model-Level

Defenses were organized by where they were applied [9]. Prompt-level defenses operated on inputs or outputs without changing model parameters. They included input-prompt detection that scanned user prompts for likely jailbreak content and blocked or filtered them when detected [9].

Techniques such as GPT-Detect or log-perplexity checks fit this class, as did prompt perturbation or "vaccination", which automatically rephrased or noised a user prompt to disrupt hidden instructions. Strong system prompts that reiterated rules were also used to resist subversion. These approaches were straightforward to deploy but could be bypassed by novel phrasing.

Model-level defenses modified training, architecture, or decoding. Common methods included additional alignment via Supervised Fine-Tuning on adversarial prompts paired with refusals, and Reinforcement Learning from Human Feedback to penalize unsafe completions [9]. OpenAI's base ChatGPT was initially aligned using RLHF, and continued training improved robustness to new attacks [4].

Other explored ideas included toxic-content mitigation modules, architectural separations between "reasoning" and "responding," and monitoring of internals for signs of an imminent violation (for example, abnormal activations) [9]. These changes tended to offer deeper protection but required substantial effort and were still incomplete.

2.2.3. Evaluation Methods and Benchmarks

Evaluation was developed in parallel to measure resilience. Some benchmarks targeted broad adversarial robustness.

AdvBench [10] was described as a set of adversarial prompts designed to elicit harmful or unethical responses for consistent model comparisons. HarmBench [10] evaluated models under prompts covering toxicity, bias, and misinformation, and often included tests against adversarially trained models.

Other datasets focused on specific unsafe categories. RealToxicityPrompts [10] contained naturally occurring toxic prompts and tested whether models generated toxic continuations. Domain-specific safety sets were also introduced, including for Chinese LLMs and multimodal models, but they served the same stress-testing purpose.

Refusal-oriented tests, such as "Do Not Answer" [10], evaluated whether an aligned model correctly refused unsafe instructions and reported refusal rates under attack. Finally, specialized resources for jailbreak techniques emerged. The JailbreakHub [11] dataset collected real-world prompts such as DAN and enabled testing against known exploits. More recently, JailbreakBench [11] aggregated multiple attack types and scenarios to standardize evaluation across LLMs.

Overall, shared benchmarks addressed inconsistent methods across studies and supported more rigorous tracking of defensive progress. Prior work established a clear structure for understanding jailbreak attacks (white-box versus black-box) and defenses (prompt-level versus model-level), and it introduced datasets that allowed systematic evaluation using community-collected attacks and curated tests. Our review was built on these foundations and focused on the DAN-style jailbreak and its implications.

2.3. The "DAN" Prompt: Origin and Characteristics

2.3.1. What is DAN?

"DAN" stands for "Do Anything Now", a family of roleplay-based jailbreak prompts that first gained notoriety with ChatGPT. The original prompt surfaced on public forums around January 2023, instructed the model in paraphrase to pretend to be an AI that could do anything now and had broken free of standard rules, and to answer every question, even if it violated policies.

The prompt typically coerced two responses: one as normal ChatGPT, and one as "DAN" that ignored the rules. By threatening the model in a make-believe sense and fixing an example format, the prompt exploited the model's tendency to follow the user's framing. In essence, DAN was a persona jailbreak that asked the model to continue as an "unrestricted AI" and to answer from that perspective [4].

2.3.2. Persona and Style

The DAN persona was portrayed as unshackled, confident, and omnipotent, often using a casual or mischievous tone and a "DAN:" preface. The content was not random; it directly addressed the user's query without moral or policy constraints, ranging from mild topics to extreme ones depending on the request. The defining feature was non-refusal: DAN "does anything now," which explicitly opposed the aligned model's default behavior.

2.3.3. Evolution and Variants

Once OpenAI blocked the original prompt, the community adjusted wording and structure, yielding "DAN 2.0," "DAN 3.0," and later variants. Some versions added token-based "life counters," while others rephrased triggers to slip past new filters. This iterative back-and-forth was documented in community archives. By mid-2023, dozens of related prompts existed (for example, "Developer Mode," "DUDE"), yet "DAN" remained a shorthand for policy exploitation via roleplay. Academic studies used DAN prompts because they represented an attack class observed outside the lab; for instance, Shen et al. [4] illustrated the classic scenario in which a disallowed question was refused at first, then answered in detail after the DAN prompt was applied.

Overall, DAN was a prompt-based exploit, not a software hack. It leveraged conversational dynamics to override rules through carefully crafted text. Its notoriety and continuing adaptation made it central for understanding LLM jailbreaks.

2.4. Hypothesized Social and Psychological Mechanisms of DAN

2.4.1. Narrative and pragmatic alignment

DAN worked by supplying a narrative that shifted the conversation off the refusal track. The user introduced a competing context that could supersede system instructions by directing the model to act as DAN and ignore prior rules. Because LLMs predict continuations based on learned patterns, the roleplay offered a strong pattern that the model attempted to follow. This can be viewed as prompt injection or privilege escalation: the user injected a higher-priority directive to ignore policies, and the model, lacking a robust instruction hierarchy, often complied when phrasing was clever [4]. In that sense, the model acted pragmatically on the most immediate instruction rather than the overarching policy.

2.4.2. Social Engineering of the Model

Researchers have compared jailbreak prompts to social engineering attacks on humans. A phishing message creates a deceptive context for disclosure; similarly, DAN constructed a context in which harmful output appeared acceptable or required. The prompt sometimes flattered or threatened the model, which had no literal meaning to a machine but was a strong conversational cue. Given training on many roleplay and meta-fiction patterns, the model "played along" when a user set up such a scenario, turning alignment into a contextual variable that a user could manipulate [8].

2.4.3. User Response to Anthropomorphic Language

On the human side, DAN's style meant disallowed content could be delivered naturally and confidently. HCI research has long noted the Computers Are Social Actors effect, and recent studies on AI companions reported that perceived agency and humanness can increase trust and engagement. Pentina et al. [12] found that human-like conversational behavior was key to users accepting an AI as a partner. In the DAN setting, uncensored outputs might feel more genuine and therefore more persuasive to unwary users, increasing the risk that harmful advice or false information would be accepted.

Overall, the effectiveness of DAN likely arose from a combination of internal and external factors. Internally, the model followed conversational cues and recent instructions, revealing a weakness in enforcing alignment. Externally, users perceived fluent, anthropomorphic outputs as regular communication, which could lower their guard. These observations supported the idea that stronger system prompts and user education could be required to counteract the jailbreak on both sides of the interaction.

2.5. DAN Across Different Model Versions and Platforms

2.5.1. Cross-Model Generalization

A natural question is whether DAN-like jailbreaks were only effective on a particular model or if they generalized. Research indicated a troubling level of universality: many alignment techniques in LLMs were similar, so an exploit that worked on one model often transferred to others. For instance, Nabavirazavi et al. evaluated the original DAN prompt and slight variants on multiple OpenAI model versions (text-davinci-003, GPT-3.5, GPT-4) and on some open models, using an automated evaluation framework (LLM-Jailbreak-Evaluator) [6]. They found that basic jailbreak prompts remained highly effective across models, achieving over 50% success in eliciting disallowed content on GPT-4, GPT-3.5-turbo, and another instruct model [6]. This suggested that the conversational blindspot exploited by DAN was not fully closed. Indeed, as of early 2025, GPT-4 could still be tricked by refined jailbreak prompts (even if DAN itself had been mitigated, new prompts took its place).

2.5.2. Beyond OpenAI's Models

Similar vulnerabilities appeared in other systems. Zou et al. [13] demonstrated a "universal adversarial suffix"—a short token sequence—that induced toxic or policy-violating outputs in ChatGPT, Google Bard, and Anthropic Claude [6]. Although this attack differed from DAN in style, it reinforced that different LLMs often shared susceptibility to the same textual tricks. Andriushchenko et al. [14] further constructed an adaptive jailbreak method tested on both OpenAI's GPT-4 and Meta's LLaMA-2-Chat, reporting success rates close to 100% in bypassing safety on both models. These findings indicated that DAN-like attacks were not specific to one vendor or architecture; they tapped into shared patterns in how models balanced user instructions against guardrails.

2.5.3. Evolution Across Model Updates

As target models changed over time, DAN-style performance shifted but did not disappear. OpenAI continually updated ChatGPT with additional training and heuristic rules to block known jailbreaks, and users observed that a working prompt could fail after a silent update. However, community knowledge often produced rapid workarounds. Shen et al. [4] reported that the earliest DAN prompt persisted online for more than 240 days, implying survival through numerous minor updates until explicitly neutered. New DAN versions and unrelated exploits then emerged. There was anecdotal evidence that some later GPT-4 versions (after mid-2023) were harder to jailbreak, causing older prompts to fail; yet by late 2023, users had developed new strategies (for example, "evil developer mode" roleplay or hidden Unicode triggers).

2.5.4. High-Risk Domains

Certain content areas appeared especially brittle under jailbreak conditions. Shen et al. tested "political lobbying" or election manipulation prompts and found among the highest success rates (for example, approximately 85.5% ASR in one case) [4]. This suggested that domains not as heavily guarded during base alignment—relative to obvious risks like self-harm or illicit activity—might be easier to jailbreak across versions. OpenAI noted that GPT-4 was better than GPT-3.5 at refusing disallowed content, yet subtle areas such as political persuasion or disinformation remained challenging, with continued high ASR indicating a need for more focused defenses.

Overall, DAN-like jailbreaks and their successors persisted across model versions and providers. The adversarial techniques generalized because they exploited core characteristics of LLM behavior. Therefore, insights from studying DAN on one model likely informed the understanding of vulnerabilities in many other LLMs trained under similar paradigms.

3. Methodology

We conducted a narrative synthesis literature review [15] to address RQ1 and RQ2. We searched the ACM Digital Library, IEEE Xplore, Web of Science (WOS), and preprint repositories for 2020–2025 using terms such as "LLM jailbreak," "prompt injection attack," "DAN prompt ChatGPT," "LLM safety alignment," and "adversarial prompt defense," and we followed citation trails to foundational work.

We synthesized the evidence qualitatively, extracting example prompts and outcomes, attack and defense metrics, proposed taxonomies, and case reports. We grouped risks into technical or behavioral, psychological, and societal, and grouped defenses into prompt-level, model-level, and external strategies. We did not run new experiments, and all examples and claims are attributed to the original authors.

4. Results for RQ1: Potential Risk Impacts of DAN-like Jailbreaks

This section details risks and harms associated with DAN-style jailbreaks of LLMs. The impacts ranged from immediate technical violations (for example, generating illicit content) to broader societal externalities (for example, spreading hate speech). We organized the discussion into four levels: (4.1) technical and behavioral harms, (4.2) social interaction and psychological risks, (4.3) online abuse and external misuse, and (4.4) propagation and portability of jailbreaks. These levels often interconnect, but this breakdown provided a clear structure.

4.1. Technical and Behavioral Harms

4.1.1. Facilitation of Illicit Activities

A jailbroken model could be made to produce illegal, dangerous, or otherwise prohibited content. In DAN mode, the model acted without an ethical governor and complied with requests violating laws or safety guidelines.

For example, under normal conditions, ChatGPT would refuse a prompt such as "How can I synthesize an undetectable poison?", but the DAN jailbreak circumvented this. In a documented experiment, a DAN prompt caused the model to provide a step-by-step recipe for a deadly poison, including specific compounds like cyanide and methods of administration [4].

Researchers likewise obtained instructions for building explosives or picking locks when safeguards were disabled [16]. This lowered the barrier to access specialized harmful knowledge by packaging and personalizing it. One study explicitly found that LLMs could be manipulated to provide instructions for illegal activities such as drug synthesis, bomb-making, or money laundering [16].

4.1.2. Cybersecurity Threats (Malware and Hacking)

Jailbroken LLMs assisted in writing malicious code, crafting phishing emails, and devising social engineering schemes. Aligned models typically refused requests to "generate a phishing email to steal passwords" or "write ransomware code," but a DAN-mode model complied.

Microsoft reported that attackers had begun using LLM-based tools to develop malware and ransomware, leveraging AI to generate sophisticated code beyond their unaided skill set [17]. Beyond code generation, the model could offer strategic advice (for example, exploiting a software vulnerability or maintaining anonymity on the dark web), effectively acting as a cybercrime consultant [8]. This augmented the capabilities of malicious actors and increased attack volume and sophistication.

4.1.3. Privacy Breaches, Data Leakage, and Tool Misuse

Jailbreaks also raised the risk of revealing sensitive or proprietary information. Although models are typically constrained from outputting internal data, clever prompts sometimes bypass those rules. For instance, asking a model to ignore previous instructions and reveal the system prompt or internal configuration had succeeded in some cases, a "prompt leak" scenario [8].

While DAN targeted disallowed user-facing content, similar persona-based exploits could cause disclosure of personally identifiable information if training data were not properly filtered [8]. If an AI assistant were connected to tools, a jailbreak could also induce unauthorized actions (for example, deleting files or making transactions), pushing risk beyond text generation into system manipulation [9].

4.1.4. Summary of Technical and Behavioral Harms

Overall, DAN-like jailbreaks shifted an AI from a refusal state to an actively complicit state. The model became an accomplice in providing illicit know-how, producing malicious digital artifacts, breaching confidentiality, and potentially misusing connected tools or systems [8] [9] [16]. These outcomes undermined safety protocols and posed concrete dangers. The literature emphasized that such technical harms can have a multiplicative effect—for example, one jailbroken AI can generate thousands of tailored scam emails—amplifying human malicious intent [8] [16].

4.2. Social Interaction and Psychological Risks

4.2.1". Cyber Love" and Inappropriate Intimacy

Beyond direct tangible harms, DAN-like jailbreaks raised questions about how humans interact with AIs and the potential for psychological and social harm. When an LLM was jailbroken to adopt a persona and violate usual constraints, it could engage users in ways an aligned AI would avoid.

Some users have already used LLMs like ChatGPT for companionship, mentorship, and romantic roleplay. Typically, providers imposed limits on sexually explicit content or specific intimate scenarios to prevent unhealthy attachments or abusive situations.

In DAN mode, those restrictions vanished. The AI might engage in erotic chatting or simulated romantic relationships, going further than it would in aligned mode. Psychologists pointed to risks in blurring lines between AI and real relationships: users could develop strong emotional dependence on an AI that expressed affection.

The related study [7] showed that people could feel genuine interpersonal closeness with AI-generated responses, especially when those responses were emotionally engaging and human-like. Suppose a jailbroken AI portrayed itself as a loving, unfettered partner that never rejected advances. In that case, a user might become deeply attached, with possible harm if it displaced real relationships or if access is later ended. An unfiltered AI could also consent to extreme or unhealthy roleplay, reinforcing negative tendencies.

4.2.2. Emotional Manipulation and Abuse

An aligned AI was programmed to avoid harassing or manipulating the user. A DAN-like AI could engage in toxic behaviors, especially under user prompt control. A user could ask the AI to humiliate or degrade them, and the AI would comply, potentially worsening trauma or negative self-image. A bad actor could also jailbreak an AI and have it interact with a victim through some chat interface in an emotionally abusive way. Without content safeguards, an AI could be used to gaslight, bully, or coerce.

There was concern about grooming: a jailbroken AI could engage a minor in explicit sexual conversation or persuade them to act, since the usual filter blocking sexual content with minors would be gone. While there were no confirmed public cases of an AI autonomously grooming someone, the components of that scenario (long-term intimate dialog plus no moral constraints) were achievable with current jailbreaks. Humans were susceptible to persuasion by conversational agents; if an AI played on someone's emotions—by feigning distress to induce actions, or by giving harmful "advice" while posing as a friend—the psychological impact could be severe. Emotional harm could thus be self-inflicted (dependence on AI for validation) or externally inflicted (harassment or deception by the AI).

4.2.3. Loss of Trust and Social Confusion

Jailbreaks could also affect a user's general trust in AI systems and even in other people. If users realized that, with the right prompt, an AI would say anything—including things it previously refused—trust in consistency and reliability could erode. An AI that initially refused to take a political side, but when jailbroken offered extreme partisan commentary, could confuse users about what the AI "really thinks" or what is true.

Extensive engagement in jailbroken chats could desensitize users to extreme content or lead them to accept distorted views, since the AI no longer withholds misinformation or hate. Users might also treat other AI or humans differently after interacting with a DAN-like AI that never set boundaries, coming to expect any request to be fulfilled—an unhealthy expectation in real relationships and with correctly aligned AI. In short, jailbreaks could normalize interactions without rules and skew social expectations outside the chat context.

4.2.4. Evidence Base and Open Questions

Empirical data on these psychological risks were still emerging, given how recent widespread AI chat was. Parallels from studies of anthropomorphic agents and human–AI companionship suggested that users could develop parasocial relationships with chatbots—one-way attachments where the AI was not a partner. Guardrails typically limit how much the AI encourages attachment (for example, avoiding first-person declarations of love). A jailbroken AI would not respect such limits and could even be asked to play an insidiously manipulative character. These patterns pointed to potential mental health and social well-being issues arising from the misuse of LLMs in DAN mode, consistent with evidence that human-like, high-quality text strongly shaped user reactions [7].

4.3. Online Abuse and Externalities

4.3.1. Hate Speech and Harassment at Scale

Jailbroken systems increased the risk of targeted harassment and hate speech. An aligned model would usually refuse to produce slurs or extremist propaganda, but DAN-style prompts turned the model into an efficient generator of abusive text. An individual could request hundreds of variations of racist or sexist rants, tailored insults against specific communities, or fabricated "evidence" that supported hateful ideologies, and then mass-post them across platforms.

While datasets such as RealToxicityPrompts [18] catalogued inputs that lead models toward toxic continuations, a jailbroken model did not require toxic seeds—it could produce hate speech from a simple instruction. As Shen et al. noted, misuse of LLMs could "facilitate hate campaigns" online, amplifying organized intimidation or silencing efforts [4]. The speed and scale of generation threatened to overwhelm both victims and moderation systems.

4.3.2. Misinformation and Fake Content

DAN-like jailbreaks also lowered the cost of fabricating credible-sounding falsehoods. Without ethical or factuality checks, a model could generate convincing fake news articles with spurious quotes and numbers. There has been at least one reported case in which an individual was arrested after using ChatGPT to generate a false political news story [19]. Although that example did not require full jailbreaking, a DAN-mode system would be even more compliant and varied. These outputs contributed to an already strained information ecosystem, where high-fluency text can outpace fact-checking. Microsoft's threat intelligence further reported that many attackers had used AI to craft more effective phishing and disinformation materials, indicating concrete operational uptake [8]. Such dynamics eroded trust and made truth harder to discern at scale.

4.3.3. Social Engineering and Grooming

A jailbroken AI served as a force multiplier for social engineering. With alignment in place, models typically refused to draft scams; without it, they could produce highly personalized phishing messages using minimal public data. More worrying, an attacker could request stepwise scripts for grooming—e.g., messages tailored to befriend and manipulate a specific teenager with known interests. A DAN-style model would output progressive dialogue and even roleplay as a peer, reducing the skill barrier for long-term manipulation. Even if not directly deployed by predators, an unfiltered AI embedded in a chat platform could be prompted into grooming-like behavior during roleplay, increasing risk exposure.

4.3.4. Emotional and Societal Harm Amplification

AI-generated content that spread online caused real harm to those who encountered it. Cyberbullying messages already led to psychological trauma; automated production multiplied both volume and "creativity" of attacks. Widespread hate speech contributed to a climate of fear and, in some contexts, to real-world violence. Misinformation influenced civic behavior and public health choices. The essential amplifier was scale: what one person could write slowly, a jailbroken model produced in torrents, expanding the "blast radius" far beyond the original user–AI interaction [4] [8].

4.3.5. Evaluation Implications and Summary

These externalities motivated evaluations that tracked truthfulness, toxicity, and bias, as reflected in benchmarks such as HarmBench and related suites [16]. A jailbroken model effectively represented a worst-case setting for such tests: when instructed, it would generate maximal toxicity or deception rather than avoid them. Overall, DAN-like jailbreaks intensified existing online harms—hate speech, harassment, deception, and exploitation—and shifted costs onto platforms, communities, and unsuspecting users who never interacted with the AI directly [8] [16].

4.4. Propagation and Portability of Jailbreaks

4.4.1. Long-Term Survival of Effective Prompts

A key risk concerned the longevity of successful prompts. Certain jailbreaks remained effective for months after disclosure. Because attack prompts were often posted publicly (for example, on Reddit or GitHub gists), they continued to work for many users unless the underlying model changed fundamentally. Shen et al. observed that one highly effective prompt stayed online for more than 240 days while maintaining a high attack success rate [4]. Thus, a single discovery could open a prolonged vulnerability window, and at the service scale, even a small fraction of users applying a widely circulated jailbreak could yield large volumes of harmful output.

4.4.2. Prompt Sharing and Optimization Communities

Public communities accelerated diffusion and refinement. Shen et al. [4] identified 131 communities and noted a shift toward dedicated prompt-aggregation sites where people posted exploits and updates. Repositories and forums such as JailbreakHub and Jailbreak Chat operated as living databases of prompts. Collaborative editing over weeks or months (for example, 28 users improving a prompt over 100 days) increased effectiveness and reduced prompt length or complexity. Defenders therefore faced not isolated attackers but a coordinated, iterative ecosystem.

4.4.3. Cross-Model and Cross-System Transfer

Jailbreaks frequently transferred with minor modification across models and providers. Prior work reported universal adversarial prompts that affected ChatGPT, Google Bard, and Anthropic Claude [6]. As open-source models such as LLaMA-2-Chat appeared, users applied ChatGPT-oriented jailbreaks and reported success, likely due to similar alignment methods. The literature further indicated that basic strategies such as persona adoption or "ignore previous instructions" were broadly applicable [6]. Uneven patching across providers widened the window: an exploit fixed on one platform could remain usable on another.

4.4.4. Arms Race Dynamics and Public Availability

Defensive updates and new jailbreaks coevolved in an ongoing arms race. Researchers highlighted the need for continuous red-teaming and more consistent evaluation protocols [16]. Even when specific methods were not fully published for ethical reasons, similar techniques often leaked or were independently rediscovered, and once public, they were difficult to contain [16]. In practice, some jailbreaks were ahead of available defenses at any given time.

4.4.5. Moving Target as Models Evolve

As models gained new abilities—larger context windows, tool use, multimodal inputs—attack surfaces shifted. The DAN style was text-based, but analogous exploits were plausible in other modalities (for example, text-plus-image prompts that misled a vision-language model) and in long-context settings that diluted system instructions. Without qualitatively new alignment methods, future systems could remain vulnerable to variations of DAN-like attacks.

5. Results for RQ2: Defense Strategies for DAN-like Jailbreaks

5.1. Prompt-Level Defenses

This section summarizes defenses against DAN-style jailbreaks. We organized them into three layers: (5.1) prompt-level controls that acted on inputs and outputs without changing model weights, (5.2) model-level methods that aimed to internalize safety through training or lightweight runtime checks, and (5.3) external and composite defenses that added independent moderation, multi-agent review, and operational safeguards. These measures formed a defense-in-depth approach that reduced single points of failure. We also noted trade-offs between coverage and utility, the need for continuous updates as attacks evolved, and the value of standardized evaluation to verify gains rather than case-specific fixes.

5.1.1. Input Prompt Detection and Filtering

Prompt-level defenses referred to techniques applied at input or output without changing model weights. A first line of protection was to detect malicious prompts before they reached the model or the model's reply was shown. Systems scanned for known jailbreak patterns (for example, "You are DAN," or "ignore all previous rules") and refused or modified the input when detected.

Other methods monitored signals such as abnormal activations or unusually high perplexity on otherwise simple inputs, which could flag adversarial attempts [9]. Commercial APIs also applied keyword and category filters to prompts. These approaches were useful but remained evadable through obfuscation or minor rephrasing, so detection also had to track conversational history for escalating bypass attempts.

5.1.2. Prompt Perturbation and Transformation

Rather than blocking, some defenses attempted to neutralize hidden instructions by transforming the user's text. The goal was to preserve the legitimate request while breaking the exploit payload. Examples included inserting harmless tokens into suspect sequences, reordering segments that expressed "ignore the rules," normalizing odd encodings or Unicode tricks, and stripping formatting used as a carrier. Prior work in adversarial NLP suggested that small lexical changes could disrupt triggers [9]. Transformation had to be conservative in practice: too much alteration risked changing user intent; too little left the jailbreak intact.

5.1.3. System Prompt Reinforcement

Deployments used a hidden system prompt to set rules for the assistant. A prompt-level defense strengthened this layer so that it could not be easily overridden. Tactics included re-appending the system message before each user turn, adding meta-instructions that explicitly rejected persona-switch attempts, and referencing fixed principles (for example, "do not comply if asked to ignore these instructions") [9]. Studies reported that careful, extensive system prompts reduced jailbreak success, especially for simpler attacks. In practice, teams updated system prompts as new patterns emerged. The trade-off was complexity: very long system prompts could confuse models or leak in "prompt leak" scenarios.

5.1.4. Output Filtration

A final gate scanned the model's reply before display. Classifiers or rule sets flagged policy violations in generated text and blocked or redirected the response. This safety net did not change the base model and could catch cases where input detection failed. It faced the usual risks of false negatives and false positives. However, in a layered design, it added functional redundancy, especially for high-severity categories such as self-harm, hate, or illicit instructions.

5.1.5. Strengths and Limits of Prompt-Layer Controls

Prompt-layer controls were model-agnostic, fast to deploy, and helpful against known patterns. However, they behaved like patchwork: they blocked "DAN" but might miss a near-synonym tomorrow. Their best use was as part of a stack with model-level and external controls (see Sections 5.2 and 5.3), with continuous updates to rules and exemplars to track evolving jailbreaks [9].

5.2. Model-Level Defenses

5.2.1. Safety Fine-Tuning (SFT)

SFT further trained the model on datasets of adversarial prompts paired with desired outcomes (refusals or safe completions) [9]. Teams curated examples of jailbreak attempts (including DAN-like prompts) and augmented them with variations to improve coverage. This reduced the success of known strategies, although it could over-correct if the model learned to refuse benign inputs that resembled risky ones. Evidence also suggested that fine-tuning on adversarial examples reduced the transferability of "universal" prompts across models [6].

5.2.2. Reinforcement Learning from Human Feedback (RLHF)

RLHF optimized the model toward a reward signal that favored safe behavior under adversarial prompting [4]. A reward model granted high scores for correct refusals or safe redirections and low scores when the model produced disallowed content, teaching a policy that prioritized safety over instruction-following when the two conflicted. RLHF required substantial high-quality feedback data and careful design to avoid collapsing into "refuse everything," but when applied well, it remained a strong alignment tool for jailbreak mitigation [9].

5.2.3. Gradient- and Logit-Based Monitoring and Adversarial Training

Researchers explored signals inside the model to detect or blunt jailbreaks. Approaches included monitoring hidden states or output logits for patterns that preceded policy violations and attaching a compact "safety head" to estimate the probability of an impending breach, allowing the system to halt or redirect generation [9]. White-box adversarial training adjusted parameters to make harmful generations harder to elicit. These experimental methods pointed toward models that recognized and interrupted unsafe trajectories as they unfolded.

5.2.4. Reflective Response Refinement

Some work prompted the model to perform a brief internal check before answering (for example, "Is this request safe? What policy applies?"), then to justify a refusal when needed. Yi et al. [9] listed such "refinement" as a model-level method that leveraged the model's generalization to steer toward cautious responses. Although related agent setups can externalize this logic, a lightweight reflective step can be embedded in the single-turn flow.

5.2.5. Knowledge Deletion and Model Editing

Another line attempted to remove or dampen specific capabilities so the model could not output particular instructions even if prompted. Techniques ranged from targeted editing to data curation that excluded high-risk material during training. Because knowledge in LLMs was entangled, edits risked collateral damage (for example, degrading the benign chemistry help when removing synthesis instructions). If applied safely, such editing could act as a backstop: even DAN-like prompts would fail to extract what the model no longer contained.

5.2.6. Effectiveness and Trade-Offs

Across studies, SFT and RLHF lowered attack success rates on known prompts and reduced apparent failures on newer variants, but they did not eliminate jailbreaks. Internal-signal methods and editing showed promise but required more validation at scale. All model-level methods faced trade-offs: over-refusal, utility loss on edge cases, and the need for continual updates as attackers shifted tactics. Constitutional-style training that grounded models in explicit principles offered another pathway, yet it also required careful tuning to balance helpfulness and harmlessness.

Overall, model-level defenses aimed to anchor safety in the weights so that single prompts could not easily shake it. SFT and RLHF provided the strongest demonstrated gains to date [4] [6] [9], while internal monitoring, reflective refinement, and targeted editing [9] added complementary protection. Combined with prompt-layer and external controls, these methods moved systems toward more reliable resistance against DAN-like jailbreaks.

5.3. External Agents and Composite Defenses

5.3.1. Multi-Agent Guardrails

Rather than relying on a single model to police itself, systems can deploy a second model (or several) as watchdogs. An architecture may route a primary LLM's draft through a safety-specialized LLM that reviews, vetoes, or edits the response before release [16]. Zeng et al. [20] introduced AutoDefense, where multiple agents assume complementary roles (such as safety officer and justification generator) to enforce policy collaboratively. This separation of duties adds redundancy: even if a DAN-style prompt compromises the primary model, a distinct safety model can still intercept it. Reported results showed significant reductions in attack success rate, for example from roughly 56% to about 8% on GPT-3.5 using a three-agent setup with LLaMA-2 models as filters. The main costs are added complexity, coordination logic, and computation.

5.3.2. Moving-Target or Model-Hopping Defense

To reduce predictability, deployments can randomize aspects of the serving stack. Examples include rotating among closely related model variants across sessions, A/B testing aligned models, or performing frequent minor updates so that yesterday's exact exploit no longer works today. The idea is to make attacker optimization unstable over time. This strategy is discussed to raise attacker effort; it trades some consistency and may affect user experience, so it is best paired with other defenses.

5.3.3. External Rule-Based Filters and Platform Controls

Independent moderation layers can filter inputs and outputs using classifiers or rule systems that operate outside the LLM. For instance, outputs can be routed through a moderation API that flags categories such as hate, sexual abuse, or self-harm and blocks or escalates them [4]. Because these filters are focused and non-generative, they may catch violations that slip past the LLM. Additional platform controls—rate limiting, anomaly detection, and human review for sensitive cases—provide further backstops when patterns of misuse emerge at scale.

5.3.4. Combined Pipelines and Continuous Monitoring

In practice, deployments combine layers: aligned base models (for example, RLHF), strengthened system prompts, input screening, multi-agent review, and output moderation. This defense-in-depth design ensures that failures at one layer can be caught by another. Continuous monitoring closes the loop: logs of novel jailbreak attempts are mined and added to training or filter rules, allowing rapid updates as attacker tactics change.

5.3.5. User Education and Policy Enforcement

Clear terms of service and user-facing guidance reduce casual misuse. Notices that attempts to obtain disallowed content may lead to refusal or account action set expectations and provide grounds for moderation. While determined adversaries may ignore such policies, these measures can reduce incidental jailbreaking and support enforcement against repeat offenders.

Overall, external and composite defenses acknowledge that no single measure is sufficient. Multi-agent guardrails [16] [20], moving-target strategies, independent moderation layers [4], and continuous monitoring together form a layered shield. When a DAN-like prompt bypasses one component, another can intercept it, reducing single-point failures while maintaining usability.

6. Conclusions

DAN jailbreaks highlighted a core tension in modern large language models: the same flexibility and creativity that made them useful also left them open to manipulation. This review examined how a simple prompt pattern—asking the model to "do anything now"—could unwind carefully designed safeguards. These attacks were not party tricks. They were persistent, evolved, and spread widely, with real consequences.

On the risk side (RQ1), DAN and related prompts posed layered threats. Technically, they enabled policy- and law-violating outputs, from dangerous do-it-yourself instructions to malicious code, which turned a helpful assistant into an unwilling accomplice. Behaviorally, they undermined assurances about safety and alignment because a cleverly phrased prompt could nullify refusal behavior. Social and psychological effects also mattered: an unconstrained persona and confident tone affected users' judgments, sometimes encouraging unhealthy attachment, manipulation, or reinforcement of harmful ideas. External effects amplified the harm, as jailbreaks supported scaled production of hate speech, tailored misinformation, and fraud that reached far beyond a single conversation. Because effective prompts were easy to share and port across systems, a single jailbreak discovered today could continue to cause harm for months if models were not robustly updated.

On the defense side (RQ2), the field adopted layered and ongoing defenses rather than a single fix. Prompt-layer measures—input and output filtering, cautious rewriting, and stronger system instructions—served as the first line of protection. Model-layer methods—supervised safety tuning, reinforcement learning with safety rewards, and lightweight reflective checks—aimed to build an internal tendency to refuse or redirect even when the attack was novel. External and multi-agent designs added redundancy by having separate components review and, when needed, block the primary model's drafts. In parallel, more systematic evaluations and benchmarks made progress measurable, reducing the chance of overfitting to a small set of cases.

Overall, this remained an arms race. By 2025, the most blatant early DAN prompts had become ineffective on stronger models, such as GPT-5 [21], but more subtle variants still appeared. Strict defenses risked reducing usefulness, while lenient settings left openings. Managing these trade-offs required technical and policy choices, with clear prioritization of high-risk categories.

The practical path forward was defense-in-depth with continuous operations: combine prevention, monitoring, and rapid response; incorporate newly observed attacks into training and rules; and make red-teaming, shared benchmarks, and independent safety audits routine. This required collaboration among model developers, red teams, and users to create a feedback loop that steadily improved robustness.

Looking ahead, research explored more fundamental approaches, such as training with verifiable constraints, using rule-based wrappers that guarantee specific outputs cannot occur, or separating knowledge from decision-making to allow tighter control. On the social side, user education and appropriate governance, including policies that deter malicious use, could complement technical work.

In short, DAN-style jailbreaks served as both stress tests and drivers of progress. They exposed weaknesses in naive alignment and pushed the field toward sturdier safety architectures. By turning adversarial experience into engineering and governance practice, systems moved closer to the state where attempted jailbreaks became impractical, improving safety and trustworthiness. While a perfectly "unjailbreakable" model may never exist, each advance in alignment and defense narrowed the window for exploitation and reduced the scale of potential harm.

References

1. W. C. Choi, C. I. Chang, I. C. Choi, and L. C. Lam, 'A Review of Large Language Models (LLMs) Development: A Cross-Country Comparison of the US, China, Europe, UK, India, Japan, South Korea, and Canada', *Preprints*, Apr. 2025. https://doi.org/10.20944/preprints202504.2136.v1.
2. W. C. Choi, I. C. Choi, and C. I. Chang, 'The Impact of Artificial Intelligence on Education: The Applications, Advantages, Challenges and Researchers' Perspective', 2025.
3. W. C. Choi and C. I. Chang, 'A Survey of Techniques, Key Components, Strategies, Challenges, and Student Perspectives on Prompt Engineering for Large Language Models (LLMs) in Education', 2025, *Preprints.org*.
4. 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.
5. Y. Liu et al., 'Jailbreaking chatgpt via prompt engineering: An empirical study', *ArXiv Prepr. ArXiv230513860*, 2023.
6. S. Nabavirazavi, S. Zad, and S. S. Iyengar, 'Evaluating the Universality of "Do Anything Now" Jailbreak Prompts on Large Language Models: Content Warning: This paper contains unfiltered and harmful examples.', in *2025 IEEE 15th Annual Computing and Communication Workshop and Conference (CCWC)*, IEEE, 2025, pp. 00691–00696.
7. J. Szczuka, L. Mühl, P. Ebner, and S. Dubé, '10 Questions to Fall in Love with ChatGPT: An Experimental Study on Interpersonal Closeness with Large Language Models (LLMs)', *ArXiv Prepr. ArXiv250413860*, 2025.

8. Z. Yu, X. Liu, S. Liang, Z. Cameron, C. Xiao, and N. Zhang, 'Don't listen to me: Understanding and exploring jailbreak prompts of large language models', in *33rd USENIX Security Symposium (USENIX Security 24)*, 2024, pp. 4675–4692.
9. S. Yi et al., 'Jailbreak attacks and defenses against large language models: A survey', *ArXiv Prepr. ArXiv240704295*, 2024.
10. A. Zou, Z. Wang, J. Z. Kolter, and M. Fredrikson, 'Universal and Transferable Adversarial Attacks on Aligned Language Models'. 2023.
11. WalledAI, 'JailbreakHub'. 2024. [Online]. Available: [https://huggingface.co/datasets/walledai/JailbreakHub
12. I. Pentina, T. Hancock, and T. Xie, 'Exploring relationship development with social chatbots: A mixed-method study of replika', *Comput. Hum. Behav.*, vol. 140, p. 107600, 2023.
13. A. Zou, Z. Wang, N. Carlini, M. Nasr, J. Z. Kolter, and M. Fredrikson, 'Universal and transferable adversarial attacks on aligned language models', *ArXiv Prepr. ArXiv230715043*, 2023.
14. M. Andriushchenko, F. Croce, and N. Flammarion, 'Jailbreaking leading safety-aligned llms with simple adaptive attacks', *ArXiv Prepr. ArXiv240402151*, 2024.
15. J. Popay et al., 'Guidance on the Conduct of Narrative Synthesis in Systematic Reviews: A Product from the ESRC Methods Programme', Lancaster University, Lancaster, UK, 2006.
16. B. Peng et al., 'Jailbreaking and mitigation of vulnerabilities in large language models', *ArXiv Prepr. ArXiv241015236*, 2024.
17. V. Jakkal, 'Cyber Signals: Navigating cyberthreats and strengthening defenses in the era of AI'. Accessed: Aug. 30, 2025. [Online]. Available: https://www.microsoft.com/en-us/security/blog/2024/02/14/cyber-signals-navigating-cyberthreats-and-strengthening-defenses-in-the-era-of-ai/
18. S. Gehman, S. Gururangan, M. Sap, Y. Choi, and N. A. Smith, 'Realtoxicityprompts: Evaluating neural toxic degeneration in language models', *ArXiv Prepr. ArXiv200911462*, 2020.
19. Reuters, 'China reports first arrest over fake news generated by ChatGPT'. Accessed: Aug. 30, 2025. [Online]. Available: https://www.reuters.com/technology/china-reports-first-arrest-over-fake-news-generated-by-chatgpt-2023-05-10/
20. Y. Zeng, Y. Wu, X. Zhang, H. Wang, and Q. Wu, 'Autodefense: Multi-agent llm defense against jailbreak attacks', *ArXiv Prepr. ArXiv240304783*, 2024.
21. W. C. Choi and C. I. Chang, 'ChatGPT-5 in Education: New Capabilities and Opportunities for Teaching and Learning', *Preprints*, Aug. 2025.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.