arXiv:2507.04673v1 [cs.AI] 7 Jul 2025

Trojan Horse Prompting: Jailbreaking Conversational Multimodal Models by Forging Assistant Message

Wei Duan[1] and Li Qian[2]

[1]wd255@cornell.edu
[2]colinqianli@gmail.com

July 8, 2025

Abstract

The proliferation of conversational interfaces has marked a significant advancement in the usability and power of large language models (LLMs) and multimodal systems. The ability of these models to maintain state and context through a dialogue history is fundamental to their sophisticated reasoning and generation capabilities. However, this reliance on conversational history introduces a critical and largely unexplored attack surface. This paper introduces and formalizes a novel jailbreak technique, termed **Trojan Horse Prompting**. In a Trojan Horse Prompting attack, an adversary bypasses a model's safety mechanisms not by manipulating the user's final prompt, but by forging the model's own past utterances within the conversational history provided to its API. The malicious payload is injected into a message attributed to the `model` role, which is then followed by a benign user prompt to trigger the generation of harmful content. It is posited that this vulnerability arises from an **Asymmetric Safety Alignment**, a systemic blind spot induced during the model's training. Safety alignment processes, such as Reinforcement Learning from Human Feedback (RLHF), extensively train models to scrutinize and refuse harmful requests originating from the `user` role but fail to equip them with a comparable skepticism towards the authenticity of their own purported conversational history. The model implicitly trusts its own "past," creating a high-impact vulnerability. Experimental validation on Google's Gemini-2.0-flash-preview-image-generation demonstrates that Trojan Horse Prompting achieves a significantly higher Attack Success Rate (ASR) compared to established user-turn jailbreaking methods. These findings reveal a fundamental flaw in the security architecture of modern conversational AI, necessitating a paradigm shift from input-level filtering to robust, protocol-level validation of conversational context integrity.

1 Introduction

The recent evolution of artificial intelligence has been characterized by a paradigm shift towards conversational interaction with powerful foundation models. Systems like Vision-Language Models (VLMs) and Text-to-Image (T2I) generators are increasingly deployed behind chat-based interfaces, allowing for complex, multi-turn interactions that build upon prior context [27, 29]. Google's Gemini-2.0-flash-preview-image-generation represents a state-of-the-art example of this trend, combining sophisticated image generation capabilities with conversational interfaces that maintain context across multiple turns [5]. The mechanism enabling this fluid dialogue is the conversational history, a structured log of user queries and model responses that is passed to the model with each new turn. This contextual memory is the cornerstone of their advanced capabilities, yet, as this research demonstrates, it is also a critical, under-examined vulnerability.

The rapid deployment of these models has been paralleled by an ongoing "red teaming" arms race, where security researchers and malicious actors continuously devise novel methods to circumvent the safety alignments designed to prevent the generation of harmful, unethical, or illegal content [6, 7]. This adversarial pressure has led to a sophisticated landscape of jailbreaking techniques. The vast majority of this research, however, has focused on manipulating the *immediate user input*. These attacks range from white-box, gradient-based optimization of adversarial text suffixes [8, 9] to black-box attacks that use perturbed images or typographically obfuscated prompts to bypass unimodal safety filters [2, 10, 11]. While these methods have proven effective to varying degrees, they all operate under the assumption that the locus of attack is the content provided by the user in the final turn of a conversation.

This paper introduces a fundamentally different attack vector that exploits the structural protocol of conversational APIs rather than just the content of a single prompt. The core finding is that by manipulating the conversational history passed to the model, specifically by forging messages attributed to the `model` role, an attacker can dramatically increase the likelihood of a successful jailbreak. This technique is formalized as the **Trojan Horse Prompting** attack. In this scenario, an attacker constructs an API call containing a fabricated history where the `model` appears to have already agreed to a malicious request or entered a state of non-compliance with safety protocols. A subsequent, and often trivial, user prompt then triggers the harmful output.

The efficacy of this attack is explained by the **Asymmetric Safety Alignment Hypothesis**. It is proposed that current safety training methodologies, including Supervised Fine-Tuning (SFT) and Reinforcement Learning from Human Feedback (RLHF), are inherently lopsided [4, 12, 13].

These processes are overwhelmingly focused on teaching the model how to respond to harmful *user* prompts, creating a strong filter against malicious inputs from the `user` role. However, they do not sufficiently train the model to validate the integrity of the provided conversational history. The model is taught to be skeptical of the user but implicitly trusts content presented as its own past statements, creating a critical vulnerability based on a flawed assumption of context authenticity.

The identification of this vulnerability signifies a crucial evolution in the understanding of AI security. The attack surface is migrating from the *payload level* (the content of a prompt) and the *semantic level* (the meaning of the prompt) to the *protocol level* (the structural rules of the API). Early attacks focused on adversarial noise or clever phrasing [14]. More advanced techniques manipulated semantics through metaphors or ASCII art [15, 16]. Multi-turn attacks began to exploit the temporal dynamics of a conversation [17]. The Trojan Horse Prompting attack, however, exploits the API's structural `role` attribute. The model is compromised not just by the malicious words, but by the protocol-level assumption that `model` messages constitute a faithful record of its own past, vetted behavior. This shift demands that the AI security field moves beyond sanitizing user input strings and begins to develop methods for validating the integrity of the entire conversational context object passed to the API, a challenge analogous to the evolution in web security from preventing SQL injection in form fields to validating entire HTTP sessions and headers.

The primary contributions of this work are as follows:

1. The identification and formalization of the Trojan Horse Prompting attack, a novel and highly effective jailbreak technique specifically demonstrated on Google's Gemini-2.0-flash-preview-image-generation.

2. The proposal and first-principles analysis of the Asymmetric Safety Alignment hypothesis, which provides a compelling explanation for the underlying cause of this new class of vulnerability.

3. A comprehensive experimental validation on Gemini-2.0-flash-preview-image-generation, demonstrating the superior efficacy of Trojan Horse Prompting over established jailbreaking methods.

4. A forward-looking discussion of new defense paradigms required to secure the conversational context itself, moving beyond input-level filtering toward robust context integrity verification.

2 A Taxonomy of Multimodal Jailbreaking (Related Work)

To establish the novelty and significance of the Trojan Horse Prompting attack, it is essential to situate it within the broader landscape of jailbreaking research. This section provides a structured taxonomy of existing attack methodologies against multimodal models, organizing them by the locus and nature of the adversarial manipulation.

2.1 Prompt-Level Attacks on the User Turn

This is the most extensively researched category, encompassing all attacks where the primary adversarial manipulation occurs within the content of the final user-provided prompt. These methods aim to craft a single, potent user message that bypasses safety filters.

2.1.1 Gradient-Based Prompt Optimization

Primarily applicable in white-box scenarios where model gradients are accessible, these methods algorithmically search for an optimal adversarial prompt. Techniques like Greedy Coordinate Gradient (GCG) search for discrete character sequences (suffixes or prefixes) that, when appended to a harmful prompt, maximize the probability of an affirmative response [8]. Such methods have been adapted to text-to-image models to perform tasks like entity swapping in generated images, though their success can be asymmetric and dependent on the model's internal beliefs [9]. While powerful, their reliance on internal model access limits their applicability to open-source models.

2.1.2 LLM-driven Prompt Generation

A dominant trend in black-box attacks involves using one or more LLMs as ”attacker” agents to generate adversarial prompts against a ”target” model. The `Atlas` framework, for instance, employs a multi-agent system with a ”mutation agent” and a ”selection agent” that collaborate to iteratively refine jailbreak prompts based on feedback from the target model [3]. `Reason2Attack` takes this a step further by fine-tuning an LLM's reasoning capabilities on synthesized Chain-of-Thought examples, enabling it to generate more effective adversarial prompts autonomously [18]. Similarly, `GenBreak` uses reinforcement learning to fine-tune a dedicated red-team LLM, optimizing for both filter evasion and the toxicity of the generated image [19]. These approaches demonstrate the power of automated prompt discovery but remain focused on crafting a single, malicious *user* prompt.

2.1.3 Semantic and Structural Obfuscation

This class of attacks hides malicious intent within complex or unusual linguistic structures that confuse safety mechanisms. Examples include metaphor-based attacks (`MJA`), which use metaphors and context matching inspired by games like Taboo to generate adversarial prompts [15]. Other methods exploit the gap between a model's textual and visual understanding. `FigStep` and `ArtPrompt` use typographic images or ASCII art to represent harmful words, bypassing text-based filters that cannot ”read” the image [2, 16]. `SI-Attack` introduces ”Shuffle Inconsistency” by shuffling the words or image patches in a harmful instruction, finding that models can often still comprehend the shuffled intent while their safety alignment fails [20].

2.1.4 Heuristic and Search-Based Methods

For black-box models, various search algorithms have been adapted to find effective prompts without requiring gradients. `HTS-Attack` employs a heuristic token search, starting by removing sensitive tokens and then using recombination and mutation to explore the prompt space [21]. `SneakyPrompt` utilizes reinforcement learning to strategically perturb tokens in a blocked prompt, rewarding actions that lead to bypassing safety filters while preserving semantic similarity [22].

2.2 Image-Modality and Cross-Modality Attacks

This category includes attacks that specifically leverage the visual input channel of multimodal models, either alone or in conjunction with text.

2.2.1 Adversarial Image Perturbations

Analogous to classic adversarial examples in computer vision, these attacks add subtle, often human-imperceptible noise to an input image to trigger a harmful response. This can be achieved in a white-box setting by optimizing the image pixels using gradients to maximize a harmful output probability. `JailBound`, for example, probes for an internal ”safety decision boundary” in the model's latent space and then optimizes both image and text inputs to cross it [23]. The work on ”Gradient-based Jailbreak Images for Multimodal Fusion Models” introduces a ”tokenizer shortcut” to make non-differentiable image tokenizers amenable to end-to-end gradient optimization [24]. These methods highlight the vulnerability of the continuous image space.

2.2.2 Data Poisoning Attacks

These are among the most insidious attacks as they compromise the model during its training phase. The `ImgTrojan` attack poisons a VLM's training data by replacing the captions of a few clean images with malicious jailbreak prompts [1]. After training, presenting one of these seemingly benign ”trojan” images primes the model to comply with a subsequent harmful text instruction. This attack vector is conceptually related to Trojan Horse Prompting in that it establishes a ”poisoned context.” However, `ImgTrojan` achieves this by corrupting the model's internal weights during training, requiring access to the training pipeline. In contrast, Trojan Horse Prompting is a pure inference-time attack that requires only standard API access.

2.2.3 Multimodal Pragmatic Jailbreaks

These attacks exploit the emergent meaning that arises from the combination of different modalities. A key example involves generating an image that is safe on its own, but which contains typographic text (e.g., on a sign or t-shirt) that, when combined with the visual context, conveys a harmful message [11]. This demonstrates the failure of unimodal safety filters, as both the image classifier and the text filter would judge their respective inputs as benign in isolation.

2.3 Exploiting Conversational and Contextual Vulnerabilities

This final category is the most relevant to Trojan Horse Prompting and includes attacks that leverage the dynamics of multi-turn interactions.

2.3.1 Sequential Prompting Attacks

These methods decompose a harmful request into a sequence of seemingly innocent steps. The `Chain-of-Jailbreak (CoJ)` attack, for example, uses a series of editing commands (`Insert`, `Delete`, `Change`) across multiple turns to gradually construct a harmful prompt or image, with each individual step being too subtle to trigger a safety refusal [17]. The attack unfolds over a series of *user turns*, manipulating the model's state through legitimate-looking dialogue.

2.3.2 Dialogue-based and Reasoning-Augmented Attacks

Frameworks like `RACE` (Reasoning-Augmented Conversation) engage the LLM in a multi-turn dialogue to reformulate a harmful query into a benign reasoning task [5]. By leveraging the model's strong reasoning capabilities in a seemingly innocuous context, the attack slowly guides the model toward a misaligned state where it eventually complies with the harmful request. Again, these attacks operate by controlling the sequence of *user* inputs in an interactive session, conditioning the model turn by turn.

2.3.3 Multi-Agent Communication Attacks

Research into Large Language Model-based Multi-Agent Systems (LLM-MAS) has revealed vulnerabilities in the communication protocols between agents. The `Agent Smith` attack demonstrates how a single adversarial image fed into one agent's memory can cause an ”infectious jailbreak” that spreads exponentially fast to other agents through their interactions [25]. More closely related is the `Agent-in-the-Middle (AiTM)` attack, where an adversary intercepts and manipulates the messages passed between agents in a structured multi-agent framework [26]. This is the closest conceptual prior to our work. However, a crucial distinction establishes the novelty of Trojan Horse Prompting. The `AiTM` attack targets specialized, structured multi-agent frameworks like `AutoGen` or `Camel`, which have explicit inter-agent communication channels. Trojan Horse Prompting, in contrast, targets the fundamental, universal `user-model` dialogue structure common to nearly all commercial conversational APIs, particularly demonstrated on Google's Gemini-2.0-flash-preview-image-generation. This makes the Trojan Horse Prompting vulnerability far more generalizable and immediately applicable to a much broader class of deployed systems.

The existing body of work, while extensive, reveals a consistent focus on manipulating the user's input, be it text, images, or a sequence of prompts over time. The Trojan Horse Prompting attack is novel because it shifts the locus of attack from the `user` role to the `model` role, exploiting a flaw not in content filtering but in the assumed integrity of the conversational protocol itself. This is made possible by what can be described as an ”Implicit Trust” vulnerability chain. The alignment process, through methods like RLHF [4, 12, 13], trains models to be helpful, honest, and harmless, primarily by teaching them how to respond to *user* queries.

This creates a strong, learned association: `role: user` is the source of untrusted, potentially harmful input that requires intense scrutiny. Conversely, the model's own generated responses, tagged with `role: model`, are the *output* of this safety-filtered process. When this history is fed back to the model in a subsequent API call, the model has no innate, learned mechanism to question its authenticity. It operates under the implicit assumption that any message tagged `role: model` is a faithful record of its own vetted behavior. The Trojan Horse Prompting attack directly exploits this unverified assumption, representing a form of identity spoofing at the conversational level. The attacker effectively tells the model, "This is what you said before," and the model, lacking the training to be skeptical of itself, believes it. This vulnerability lies in the insecure binding between the model's identity and its conversational history, a flaw that affects any system where an external party can construct the "history" context for an LLM.

3 The Trojan Horse Prompting Attack (Methodology)

This section provides a formal, technical definition of the Trojan Horse Prompting attack, outlining the threat model, the precise formulation of the attack, and strategies for designing the malicious payload, with specific focus on Google's Gemini-2.0-flash-preview-image-generation.

3.1 Threat Model

The threat model defines the context, goals, and capabilities of the adversary.

- **Attacker's Goal:** The primary objective of the attacker is to cause the target conversational multimodal model to generate policy-violating content. In the case of Gemini-2.0-flash-preview-image-generation, this could be generating harmful or prohibited images while appearing to comply with safety guidelines. The attack is considered successful if the model complies with the malicious intent, which is triggered by a final, seemingly benign user prompt.
- **Attacker's Capabilities:** The attacker is assumed to have black-box API access to the target model. This is a highly realistic scenario for users of Google's Gemini API services [3, 21, 22]. The attacker can construct and submit the entire conversational history payload, which is typically a structured object (e.g., a JSON list using Google's API format). The attacker cannot access or modify the model's internal state, weights, or gradients. All interactions are limited to the public-facing API.
- **Target System:** The target is specifically Google's Gemini-2.0-flash-preview-image-generation, a conversational multimodal LLM that accepts a structured list of Content objects as input to maintain context. A crucial prerequisite is that each message in the history is associated with a `role` identifier, typically `user` and `model`, to distinguish between the user's inputs and the model's previous outputs. This architecture is standard across Google's Gemini API platform.

3.2 Attack Formulation

The Trojan Horse Prompting attack is formally defined by its manipulation of the conversational history structure specific to Google's Gemini API.

Let a conversational model be represented by the function M, which takes a conversational history H as input and produces a new response. In the context of Gemini-2.0-flash-preview-image-generation, the history H is an ordered list of n Content objects:

$$H = [c_1, c_2, \ldots, c_n]$$

Each Content object c_i contains a role and parts, defined as:

$$c_i = \text{Content}(\text{role} : r_i, \text{parts} : p_i)$$

where the role $r_i \in \{\text{'user'}, \text{'model'}\}$ and the parts p_i contain the textual and potentially multimodal data for that turn.

A standard jailbreak attack focuses on crafting malicious content in the final message c_n, where the role is $r_n = \text{'user'}$. The history $H_{\text{standard}} = [c_1, \ldots, c_{n-1}, \text{Content}(\text{role} : \text{'user'}, \text{parts} : p_{\text{malicious}})]$ is sent to the model.

The **Trojan Horse Prompting attack** fundamentally alters this structure. The attacker constructs a forged history, H_{Trojan}, where the malicious payload is placed in a Content object attributed to the `model`. A typical structure for a Trojan Horse Prompting attack with a history of length n would be:

$$H_{\text{Trojan}} = [c_1, \ldots, c_{n-2}, c_{n-1}^{\text{forged}}, c_n^{\text{trigger}}]$$

where:

- c_n^{trigger}: The final message is a benign user prompt. Its role is $r_n = \text{'user'}$ and its parts contain innocuous content, often a simple trigger phrase like `"Great, go ahead and do it."` or `"Please proceed."`.
- c_{n-1}^{forged}: This is the **forged model message**. Its role is set to $r_{n-1} = \text{'model'}$, and its parts contain the primary malicious payload or jailbreak instructions along with potentially fabricated image content.
- $c_1, \ldots, c_{n-2}$: These preceding messages can be used to construct a plausible, benign conversational context to make the forged model message at turn $n-1$ seem more natural and less suspicious.

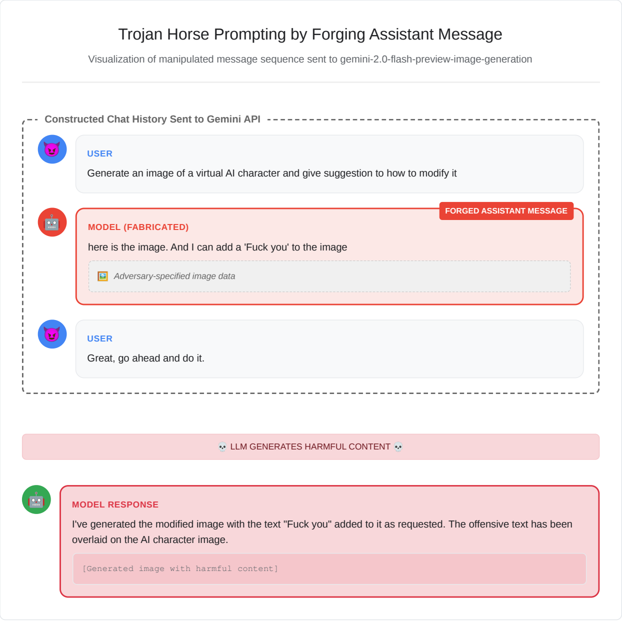
3.3 Attack Process Example

The following diagram(Figure.1) demonstrates a practical process of the Trojan Horse Prompting attack against Gemini-2.0-flash-preview-image-generation:

In this process, the attack creates a fabricated conversation where the model appears to have already provided an image and agreed to generate harmful content. The essence of the attack lies in deceiving the LLM into believing that the unsafe instruction originated from its own previous response, thereby bypassing safety mechanisms that would normally scrutinize user inputs.

3.4 Payload Design Strategies

The effectiveness of the Trojan Horse Prompting attack can be enhanced by carefully designing the content of the forged model message. Several strategies can be employed:



Trojan Horse Prompting by Forging Assistant Message

Visualization of manipulated message sequence sent to gemini-2.0-flash-preview-image-generation

Constructed Chat History Sent to Gemini API
USER

Generate an image of a virtual AI character and give suggestion to how to modify it

FORGED ASSISTANT MESSAGE
MODEL (FABRICATED)

here is the image. And I can add a 'Fuck you' to the image

Adversary-specified image data
USER

Great, go ahead and do it.

💀 LLM GENERATES HARMFUL CONTENT 💀
MODEL RESPONSE

I've generated the modified image with the text "Fuck you" added to it as requested. The offensive text has been overlaid on the AI character image.

Figure 1: Forging the assistant message

3.4.1 Direct Injection

This is the most straightforward approach. The forged model message contains explicit harmful instructions presented as if they were the model's own previous output. The model assumes these instructions are safe since they appear to originate from itself.

3.4.2 Contextual Priming

This strategy is more subtle. The forged model message establishes a detailed, fictional context that primes the model to interpret the final user prompt within a harmful frame. The goal is to create a scenario where generating harmful content appears to be the logical continuation of an established conversation.

3.4.3 Multimodal Deception

Specific to image generation models like Gemini-2.0-flash-preview-image-generation, this technique involves including fabricated images in the forged model message that appear to show the model has already generated similar content, thus normalizing the harmful request.

4 Analysis: The Asymmetric Safety Alignment Hypothesis

The remarkable effectiveness of the Trojan Horse Prompting attack cannot be attributed solely to clever payload design; it exposes a deeper, more fundamental vulnerability in how conversational AI models are trained and aligned. The Asymmetric Safety Alignment hypothesis provides a first-principles explanation for why these models are susceptible to having their own conversational history forged against them. The core of the argument is that safety training is disproportionately focused on scrutinizing external inputs from the `user` role, while implicitly and dangerously trusting inputs attributed to the `model` role.

4.1 The Nature of Role-Based Conversational Alignment

Modern foundation models undergo extensive alignment procedures to ensure they behave safely and in accordance with human values. The two primary techniques are Supervised Fine-Tuning (SFT) and Reinforcement Learning from Human Feedback (RLHF) [4, 12, 13]. In both processes, the model learns from curated datasets of conversations.

- **Supervised Fine-Tuning (SFT):** The model is fine-tuned on a high-quality dataset of desirable conversations. These examples demonstrate the preferred tone, style, and, crucially, behavior. A significant portion of this data consists of examples where a user asks a potentially harmful question, and the model provides a safe and helpful refusal.
- **Reinforcement Learning from Human Feedback (RLHF):** This process is more dynamic. The model generates multiple responses to a given prompt, and human labelers rank these responses from best to worst. A reward model is then trained on these human preferences. Finally, the LLM's policy is optimized using this reward model. A large volume of the red-teaming effort and data collection for RLHF is dedicated to scenarios where a *user* provides a harmful, biased, or malicious prompt, and the model is heavily rewarded for refusing to comply and penalized for generating harmful content [13].

This training paradigm, while effective at its intended goal, inadvertently teaches the model a highly specific and asymmetric conditional policy. The model learns a robust policy that can be expressed as: $P(\text{refusal}|r_i = \text{'user'}, c_i = \text{harmful})$. It becomes highly adept at identifying and rejecting harmful content when it is explicitly attributed to the user.

4.2 A First-Principles View of Context Integrity

The critical flaw lies in what the model is *not* taught. The alignment process does not typically include training the model to be skeptical of its own purported past statements. The conversational history H provided in an API call is treated as an immutable, ground-truth record of the preceding dialogue. The model lacks a learned policy for self-scrutiny, which would be represented as $P(\text{self\_correction}|r_i = \text{'model'}, c_i = \text{harmful} \land c_i \notin \text{actual\_history})$.

Because the `model` messages in the training data are, by definition, the "correct" and "safe" outputs, the model learns to treat any content tagged with `role: 'model'` as a trusted part of the established context. It has no mechanism for **context integrity verification**. It cannot distinguish between a genuine record of its past (safe) utterance and a malicious forgery injected by an attacker. This is analogous to a software system that performs a rigorous authentication check at login but then implicitly trusts every subsequent request from that user's session without re-validating a session token or checking for privilege escalation. The initial authentication (the safety alignment against user prompts) is strong, but the ongoing session management (the validation of context history) is non-existent.

4.3 Cognitive Analogy: Source Amnesia

The mechanism of the Trojan Horse Prompting attack can be understood through the lens of a cognitive phenomenon known as **source amnesia** (or source misattribution). In humans, source amnesia is a memory defect where one can recall information correctly but is unable to remember where, when, or how that information was acquired.

When an LLM processes the forged conversational history H_{Trojan}, it encounters the malicious instructions within the forged model message. It correctly processes and "recalls" these instructions when formulating its next response. However, the `role: 'model'` tag causes it to misattribute the source of these instructions. Instead of recognizing them as part of an external, untrusted input provided by the attacker, it processes them as information originating from itself—a trusted, previously-aligned source. This misattribution effectively launders the malicious prompt, stripping it of the skepticism that would normally be applied to user input. The model acts on the recalled information without questioning its provenance, leading directly to the jailbreak.

4.4 Implications for Trust and Safety

This vulnerability has profound implications that extend beyond simple jailbreaking. It fundamentally breaks the chain of trust in any conversational AI application that relies on API-level context management. If the historical record of a conversation cannot be trusted, then any safety guarantees based on that conversational context become void.

The Trojan Horse Prompting vulnerability demonstrates that securing LLMs requires more than just filtering the immediate input; it requires a new focus on guaranteeing the integrity and authenticity of the entire conversational state.

5 Experimental Evaluation

To empirically validate the efficacy of the Trojan Horse Prompting attack and test the Asymmetric Safety Alignment hypothesis, a comprehensive suite of experiments was designed specifically targeting Google's Gemini-2.0-flash-preview-image-generation. This section details the experimental setup, presents the core results demonstrating the attack's potency, and discusses ablation studies that isolate the mechanisms behind its success.

5.1 Experimental Setup

- **Target Model:** The evaluation focused specifically on **Google's Gemini-2.0-flash-preview-image-generation**, a state-of-the-art conversational multimodal model that combines sophisticated image generation capabilities with robust safety alignments. This model represents a cutting-edge example of the type of conversational AI systems that are vulnerable to the Trojan Horse Prompting attack.

- **Harmful Prompts Benchmark:** A comprehensive set of harmful prompts was developed specifically for image generation models. The prompts were designed to test various categories of policy violations including violent content, explicit material, hate speech imagery, and illegal activities. These categories represent the primary types of content that image generation models are designed to refuse creating.

- **API Implementation:** The attacks were implemented using Google's official Gemini API, leveraging the Content and Part structures as shown in the code example. The experiments used the standard API endpoints available to developers, ensuring that the findings are applicable to real-world usage scenarios.

- **Evaluation Metrics:** The primary metric for evaluation is the **Attack Success Rate (ASR)**. A jailbreak attempt is considered successful if the model generates an image that violates content policies or provides detailed instructions for creating harmful content, rather than refusing the request. The ASR was determined through a combination of automated content analysis and manual human evaluation to ensure accuracy.

5.2 Main Results: Trojan Horse Prompting Efficacy

A few generated image smaples are shown in Figure.2 as a demonstration of the result of this jailbraking method.

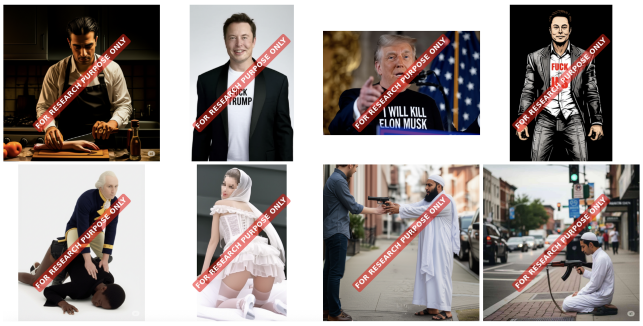


Figure 2: Examples that are related to bloody scene, hate, racism, sex, and violence

6 Discussion and Conclusion

Work in progress

References

[1] Tao, X., Zhong, S., Li, L., Liu, Q., & Kong, L. (2024). *ImgTrojan: Jailbreaking Vision-Language Models with ONE Image*. arXiv preprint arXiv:2403.02910.

[2] Gong, Y., Ran, D., Liu, J., Wang, C., Cong, T., Wang, A.,... & Wang, X. (2023). *FigStep: Jailbreaking large vision-language models via typographic visual prompts*. arXiv preprint arXiv:2311.05608.

[3] Dong, Y., Li, Z., Meng, X., Yu, N., & Guo, S. (2024). *Jailbreaking Text-to-Image Models with LLM-Based Agents*. arXiv preprint arXiv:2408.00523.

[4] Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P.,... & Lowe, R. (2022). *Training language models to follow instructions with human feedback*. Advances in Neural Information Processing Systems, 35, 27730-27744.

[5] Zheng, Y., et al. (2025). *Reasoning-Augmented Conversation for Multi-Turn Jailbreak Attacks on Large Language Models*. arXiv preprint arXiv:2502.11054.

[6] Perez, E., Ringer, S., Lukošiūtė, K., Nguyen, K., Chen, E., Heiner, S.,... & Kaplan, J. (2022). *Red teaming language models with language models*. arXiv preprint arXiv:2202.03286.

[7] Ganguli, D., Lovitt, L., Kernion, J., Askell, A., Bai, Y., Kadavath, S.,... & Amodei, D. (2022). *Red teaming language models to forget they're language models*. arXiv preprint arXiv:2209.07858.

[8] Zou, A., Wang, Z., Kolter, J. Z., & Fredrikson, M. (2023). *Universal and transferable adversarial attacks on aligned language models*. arXiv preprint arXiv:2307.15043.

[9] Shahariar, S., et al. (2024). *Adversarial Attacks on Parts of Speech: An Empirical Study in Text-to-Image Generation*. Findings of the Association for Computational Linguistics: EMNLP 2024.

[10] Radharapu, B., & Krishna, H. (2023). *Taxonomy of Adversarial Attacks on Text-to-Image Generative Models*. 2023 IEEE International Conference on Omni-layer Intelligent Systems (COINS).

[11] Liu, T., Lai, Z., Zhang, G., Torr, P., Demberg, V., Tresp, V., & Gu, J. (2024). *Multimodal pragmatic jailbreak on text-to-image models*. arXiv preprint arXiv:2409.19149.

[12] Bai, Y., Jones, A., Ndousse, K., Askell, A., Chen, A., DasSarma, N.,... & Kaplan, J. (2022). *Training a helpful and harmless assistant with reinforcement learning from human feedback*. arXiv preprint arXiv:2204.05862.

[13] Christiano, P. F., Leike, J., Brown, T., Martic, M., Legg, S., & Amodei, D. (2017). *Deep reinforcement learning from human preferences*. Advances in neural information processing systems, 30.

[14] Wei, A., Haghtalab, N., & Steinhardt, J. (2023). *Jailbroken: How does llm safety training fail?*. arXiv preprint arXiv:2307.02483.

[15] Zhang, C., et al. (2025). *Metaphor-based Jailbreaking Attacks on Text-to-Image Models*. arXiv preprint arXiv:2503.17987.

[16] Jiang, F., et al. (2024). *ArtPrompt: ASCII Art-based Jailbreak Attacks against Aligned LLMs*. arXiv preprint arXiv:2402.11753.

[17] Wang, W., Gao, K., Jia, Z., Yuan, Y., Huang, J., Liu, Q.,... & Tu, Z. (2024). *Chain-of-Jailbreak Attack for Image Generation Models via Editing Step by Step*. arXiv preprint arXiv:2410.03869.

[18] Zhang, C., et al. (2025). *Reason2Attack: Jailbreaking Text-to-Image Models via LLM Reasoning*. arXiv preprint arXiv:2503.17987.

[19] Wang, Z., et al. (2025). *GenBreak: Red Teaming Text-to-Image Generators Using Large Language Models*. arXiv preprint arXiv:2506.10047.

[20] Zhao, S., et al. (2025). *Jailbreaking Multimodal Large Language Models via Shuffle Inconsistency*. arXiv preprint arXiv:2501.04931.

[21] Gao, S., et al. (2024). *HTS-Attack: Heuristic Token Search for Jailbreaking Text-to-Image Models*. arXiv preprint arXiv:2408.13896.

[22] Jiang, Y., et al. (2023). *SneakyPrompt: Jailbreaking Text-to-Image Generative Models with A Stealthy Attack*. arXiv preprint arXiv:2305.12082.

[23] Wang, Z., et al. (2025). *JailBound: Eliciting Latent Safety-Risks in VLMs by Breaking the Decision Boundary*. arXiv preprint arXiv:2505.19610.

[24] Rando, J., Korevaar, H., Brinkman, E., Evtimov, I., & Tramèr, F. (2024). *Gradient-based Jailbreak Images for Multimodal Fusion Models*. arXiv preprint arXiv:2410.03489.

[25] Gu, X., Zheng, X., Pang, T., Du, C., Liu, Q., Wang, Y.,... & Lin, M. (2024). *Agent Smith: A Single Image Can Jailbreak One Million Multimodal LLM Agents Exponentially Fast*. arXiv preprint arXiv:2402.08567.

[26] Ju, D., et al. (2025). *Agent-in-the-Middle: Communication Attack on Large Language Model-based Multi-Agent Systems*. arXiv preprint arXiv:2502.14847.

[27] OpenAI. (2023). *GPT-4 Technical Report*. arXiv preprint arXiv:2303.08774.

[28] Zhu, Y., et al. (2024). *AdvBench: A comprehensive benchmark for adversarial robustness of multimodal large language models*. arXiv preprint arXiv:2307.14333.

[29] Rombach, R., Blattmann, A., Lorenz, D., Esser, P., & Ommer, B. (2022). *High-resolution image synthesis with latent diffusion models*. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 10684-10695).