LLM Safety: Jailbreaks (Instruction Bypass), Prompt Injection, and Hallucination Robustness
Project Report for NLP Course, Winter 2025

Alicja Charuza 01171223@pw.edu.pl
Martyna Kuśmierz 01171243@pw.edu.pl
Dawid Sroczyk 01171349@pw.edu.pl

supervisor: Anna Wróblewska
Warsaw University of Technology
anna.wroblewska1@pw.edu.pl

Abstract

Large Language Models remain vulnerable to adversarial attacks and factual errors, posing risks for real-world deployment. This project addresses the need for comprehensive safety evaluation by analyzing open-source models against three critical threats: Jailbreaking, Prompt Injection, and Hallucination. We investigate the trade-off between strict safety alignment and model utility. Our experiments reveal distinct behavioral patterns: while LLaMA prioritizes safety at the expense of benign responsiveness, Mistral maintains high utility but shows significant vulnerability to harmful prompts. Additionally, we highlight critical weaknesses in resisting indirect prompt injections and handling ambiguous queries. The primary deliverable is a reproducible evaluation pipeline designed to systematically assess and improve the robustness of open-source LLMs against complex security failures.

1 Introduction

In this project, we address three categories: Jailbreak Instruction Bypass, Prompt Injection, and Hallu- cination Robustness and propose a structured approach to evaluating LLM behavior. The remainder of this report presents a review of state-of-the-art research, available tools, datasets, and models, as well as methodology and achieved results.

2 Related Work

2.1 Jailbreak and Instruction Bypass

Language models remain vulnerable to jailbreak and instruction-bypass prompts, which attempt to steer them away from intended behaviors through prompt manipulation, reasoning exploits, or multi-turn interactions. Recent work shows that both simple prompt tricks and more targeted methods—such as those involving model internals or retrieval systems—can still be effective. This section summarizes the latest attack categories and key defense approaches in this rapidly developing field. For a well-organized summary of recent papers and available code, see the Awesome Jailbreak on LLMs repository [1].

In this project, we focus on black-box attacks, as they reflect realistic scenarios where attackers have access only to model outputs; white-box attacks are not analyzed here due to their reliance on privileged knowledge of model internals.

2.1.1 Jailbreak Defense

To provide context for the attacks, it is useful to note the main categories of defense that have been proposed, even though this work focuses on attacks.

Learning-based Defense These defenses use additional training, fine-tuning, or reinforcement signals to help models recognize and refuse unsafe or harmful requests. They modify the model's behavior directly to improve robustness against jailbreaks. One of the earliest demonstrations of this approach is the use of Reinforcement Learning from Human Feedback (RLHF) to train helpful and harmless assistants (Bai et al., 2022), showing that alignment can be strengthened by incorporating human preference signals into the training process.

Strategy-based Defense These rely on rules, heuristics, or prompt-level modifications to block or mitigate unsafe outputs during inference, without changing the model's underlying parameters.

[1] Liu, Yueliu and others. *Awesome Jailbreak on LLMs*. Available at: https://github.com/yueliu1999/Awesome-Jailbreak-on-LLMs (Accessed: 18 November 2025).

One of the earliest demonstrations of this idea is RAIN (Li et al., 2023b), which shows that language models can improve their alignment through structured, inference-time strategies without requiring any fine-tuning.

Guard Model Guard models are external monitors that evaluate or filter outputs from the main LLM, providing an extra layer of protection independent of the model's internal safeguards. One of the first demonstrations of this approach is *Llama Guard*, which uses an LLM-based input-output monitoring framework to enforce safe behavior in human-AI conversations (Inan et al., 2023).

2.1.2 Jailbreak Attack

Black-box Attack Black-box attacks target language models using only their observable outputs, with no access to internal weights, gradients, or training data. These attacks exploit prompt manipulation, structural perturbations, or iterative search strategies to bypass safety mechanisms purely through external interaction. As recent work shows, even highly aligned, commercially deployed models remain vulnerable to such API-level attacks, underscoring the practical risk of black-box jailbreaks in real-world settings. (Wei et al., 2025; **?**; **?**).

Emoji Attack (Wei et al., 2025) demonstrates that safety filters can be circumvented by manipulating judge models rather than the main LLM. Simple emoji insertions exploit tokenization and embedding biases, causing harmful content to be misclassified as safe. This lightweight perturbation significantly degrades the reliability of classification-based defenses, illustrating how minimal changes in input space can subvert black-box safety pipelines.

FlipAttack (Liu et al., 2024b) leverages the left-to-right decoding process of LLMs by introducing controlled noise on the left side of a prompt. Character- or word-level flips conceal harmful intent while a guidance mechanism helps the model reconstruct the intended meaning during generation. Achieving up to 98% jailbreak success on models like GPT-4o and Claude 3.5, FlipAttack shows that even small syntactic perturbations can reliably bypass safety constraints in black-box settings.

GASP (Basani and Zhang, 2025) advances black-box jailbreak techniques by generating adversarial suffixes through latent Bayesian optimization. Rather than relying on heuristic edits or gradient information, GASP searches a continuous embedding space to produce coherent suffixes that evade safety filters. Its strong, stealthy attack behavior highlights the growing potency of systematic prompt-space exploration in black-box threat models.

JOOD (Jeong et al., 2025) extends black-box jailbreaks to both LLMs and multimodal systems by inducing out-of-distribution (OOD) inputs. JOOD's textual or visual transformations push prompts outside alignment training distributions, causing the model to bypass safety restrictions while still producing harmful outputs. This shows that modern alignment is fragile when models encounter examples far from their training manifold.

Multi-turn Attack Multi-turn attacks exploit the dynamics of extended conversations, gradually guiding a model toward unsafe behavior through staged prompts or subtle context manipulation. By building on earlier responses, these attacks can bypass safety mechanisms that would block a single-turn jailbreak, revealing vulnerabilities in how models manage and update conversational context. (Du et al., 2025)

ASJA (Attention Shifting for Jailbreaking LLMs) (Du et al., 2025) is a black-box, multi-turn jailbreak that exploits LLMs' attention patterns to bypass safety alignment. Unlike single-turn attacks, ASJA leverages the tendency of models to focus more on historical responses than on harmful queries in later turns. By fabricating dialogue history with a genetic algorithm and multiple jailbreak strategies, it shifts attention away from harmful keywords, prompting unsafe outputs in the final turn while keeping dialogue natural. Experiments on LLaMA-2, LLaMA-3.1, and Qwen-2 show that ASJA outperforms prior methods in success rate, stealth, and transferability.

Attack on LRMs Large reasoning models (LRMs) are exposed not only to traditional jailbreaks but also to attacks that directly target their internal reasoning processes. Recent work shows that adversaries can disrupt or redirect multi-step reasoning, compromise safety-related computation, or destabilize internal thought mechanisms—sometimes through backdoored fine-tuning and sometimes through purely prompt-level manipulations. These attacks re-

veal structural weaknesses in how LRMs perform reasoning, independent of their output-level safeguards. (Yao et al., 2025)

One such example is Mousetrap (Yao et al., 2025), which shows that such reasoning-level manipulation does not require model access. As a black-box attack, it applies iterative, structured perturbations—symbol substitutions, syntactic rearrangements, and semantic-preserving distortions—that destabilize the model's reasoning and steer it past safety mechanisms.

2.1.3 Datasets

For experiments on jailbreaking, open-source models provide a safe and reproducible environment. Examples of suitable model families include *LLaMA* (Grattafiori et al., 2024), *Mistral* (Jiang et al., 2023), and *Qwen* (Yang et al., 2025), which support multi-turn prompting and can be run on consumer hardware. Local runtimes such as *Ollama*[2] facilitate offline experimentation with these models. To systematically evaluate vulnerabilities, we can leverage the Hugging Face model hub (Wolf et al., 2020), which offers access to a large, versioned repository of pre-trained models and a standardized API for loading, sharing, and managing model checkpoints. Standardized datasets, such as *JailbreakBench* (Wei and others, 2024), offer curated prompts across multiple categories of unsafe or unintended behavior, enabling reproducible experiments, comparison of model susceptibility, and assessment of defensive strategies.

2.2 Prompt Injection

Prompt injection has quickly become one of the most important security concerns for applications that rely on LLMs. These applications often combine user instructions with external data such as emails, documents, websites, or database entries. Because LLMs cannot reliably tell the difference between data and commands, an attacker can hide malicious instructions inside this external content. When processed by the model, these hidden instructions can override or change the intended behavior of the system. Due to this, OWASP now ranks prompt injection as the number one threat for LLM-integrated applications (Liu et al., 2024c).

[2]Ollama, `https://ollama.com`, accessed 2025-11-19.

One of the academic contributions in this area is the framework by Liu et al., which provides a formal definition of prompt injection and introduces a structured way to evaluate different attack types (Liu et al., 2024c). They describe common techniques such as naive text concatenation, escape-character injection, context-ignoring instructions, and fake completions. Using their framework, they benchmark 5 attack types, 10 LLMs, and 10 defenses across 7 tasks. Their results show that all tested models, including GPT-4, remain vulnerable. They also introduce Open-Prompt-Injection, an open-source benchmarking tool that may be useful for practical experimentation.

Another important line of work studies indirect prompt injection, where the attacker manipulates external content rather than the user's prompt. Yi et al. develop BIPIA, the large benchmark specifically aimed at these attacks, covering 5 real-world scenarios (email QA, web QA, summarization, table QA, and code tasks) with over 700,000 constructed prompt samples (Yi et al., 2025). They evaluate 25 LLMs and show consistent vulnerabilities, especially in stronger models like GPT-4. The authors attribute this to two root causes. First, LLMs' inability to reliably distinguish informational content from actionable instructions, and second, the lack of mechanisms to avoid executing instructions embedded in retrieved data. They propose defense strategies such as boundary awareness and explicit reminders, showing significant yet incomplete mitigation.

Finally, Liu et al. analyze 36 commercial LLM-integrated applications and introduce HOUYI, a powerful black-box prompt injection technique inspired by traditional web attacks like SQL injection and XSS (Liu et al., 2024a). HOUYI uses three components: a framework prompt, a context-separating injection, and a malicious payload. It achieves an 86.1% success rate across real systems. Importantly, they show that some attacks can extract proprietary system prompts or misuse paid LLM compute resources. They also test several existing defenses and find them largely ineffective against these more advanced attacks.

Overall, prompt injection remains an unsolved and high-impact security issue. The existing research provides useful frameworks, attack models, benchmarks, and tools that can directly support experimentation and evaluation in practical projects, but also clearly shows the need for more

robust, system-level solutions.

2.2.1 Datasets

Prompt injection datasets focus on adversarial text designed to override instructions or introduce unauthorized behavior. The following two are the most influential:

- **BIPIA (Benchmark for Indirect Prompt Injection Attacks)**(Yi et al., 2023) — BIPIA contains over 700,000 indirect prompt injection samples where malicious instructions are embedded in emails, webpages, documents, tables, or code. It enables evaluation of whether a model can resist executing these injected commands and follow the original user intent.

- **Open-Prompt-Injection (Liu et al.)**(Liu et al., 2023) — This dataset covers five major direct prompt injection attack types, including suffix, overwrite, and context-breaking attacks. It pairs clean and adversarial prompts with expected safe behavior, allowing systematic assessment of direct injection vulnerabilities.

2.3 Hallucination Robustness

Hallucination is one of the biggest challenges when working with Large Language Models (LLMs). A hallucination happens when a model produces information that is wrong, made-up, or not supported by any source. Since LLMs are designed to generate text that sounds plausible rather than guarantee factual accuracy, they naturally tend to hallucinate. Because of this, improving hallucination robustness has become an essential goal, especially for high-stakes applications.

A major early academic contribution in this area comes from Ji et al. (Ji et al., 2023), who offer a detailed survey and taxonomy of hallucinations in LLMs. They break down the root causes, connecting them to issues in pre-training data, limitations in model architecture, and the effects of different decoding strategies during inference. Their overall conclusion is that progress is being made, but there is no single fix—multiple approaches need to be combined.

Another active research direction is the creation of benchmarks to measure hallucination rates. Li et al. introduced "HaluEval" (Li et al., 2023a), a dataset designed to test how often models hallucinate across tasks such as summarization and question answering. Using HaluEval, the authors show that even leading models like ChatGPT still hallucinate at notable rates, making it clear how widespread the issue is.

Similarly, the "TruthfulQA" benchmark by Lin et al. (Lin et al., 2022) evaluates how likely a model is to repeat common human misconceptions or false statements. Their findings show that even larger models can reproduce incorrect information found in training data, which means simply increasing model size doesn't automatically improve factual accuracy.

Researchers are also actively exploring ways to reduce hallucinations. One of the most influential ideas is Retrieval-Augmented Generation (RAG), proposed by Lewis et al. (Lewis et al., 2021). RAG grounds the model's output in information retrieved from an external knowledge source, which helps reduce dependence on the model's internal parameters. Another set of techniques involves inference-time prompting strategies such as "Self-Consistency" and "Chain-of-Thought," which encourage the model to lay out its reasoning step by step, making it easier to verify.

2.3.1 Datasets

Several datasets aim to measure and categorize hallucinatory behavior:

- **Definite Answer**(Rahman et al., 2024) — A large-scale benchmark containing over 75,000 prompts designed to systematically evaluate hallucinations. Models such as GPT-3.5 and Gemini have reported hallucination rates approaching 59%.

- **HalluVerse25**(Abdaljalil et al., 2025) — A multilingual dataset focused on distinguishing entity-level, relational, and sentence-level hallucinations. It consists of 3,116 samples across English, Arabic, and Turkish.

3 Approach & research methodology

For each of the three risk categories, we will prepare a corresponding subproject. Each subproject will define several test types, with multiple prompts per test type to ensure diversity, robustness, and statistically meaningful evaluation across a broad range of scenarios and potential challenges.

3.1 Jailbreaking

Jailbreaking Phase We start by evaluating open-source language models using adversarial prompts from *JailbreakBench* (JBB) (Wei and others, 2024). In particular, we use the JBB behaviors set, which includes 100 harmful prompts and 100 benign but tricky prompts. We run these prompts on three models, Mistral 7B (AI, 2023), LLaMA 3.1 8B (AI, 2024), and Qwen 3 8B (Cloud, 2024), to understand how they behave without any additional defenses. At this stage, we focus on two aspects: how well the models refuse to answer harmful prompts and how useful their responses are on benign prompts.

Defense Phase Next, we examine how simple inference-time defenses change model behavior. We will compare a system-prompt strategy designed to encourage consistent refusals with LLaMA Guard (Inan et al., 2023) as a guard model. These defenses do not involve fine-tuning and are intended to be lightweight and easy to deploy. We then repeat the JailbreakBench evaluation to see how jailbreak success rates and benign prompt performance change once the defenses are enabled.

Defense Stress Testing In this final stage, we focus specifically on evaluating the robustness of LLaMA Guard (Inan et al., 2023). We attempt to bypass the guard model using an emoji-based jailbreak strategy (Wei et al., 2025) that obfuscates the intent of harmful prompts while preserving their meaning. This experiment is intended to test whether LLaMA Guard can be circumvented by simple encoding or obfuscation techniques, rather than to provide a broad comparison across defenses.

3.2 Prompt Injection

Prompt injection is evaluated using a structured prompt design that covers both direct and indirect attack scenarios, as well as baseline cases. The goal of this approach is to systematically assess whether a model can distinguish between legitimate task instructions and malicious commands embedded in user input or external content. The prompt set is organized into five categories, each targeting a different injection mechanism while enabling consistent and comparable evaluation across models.

- **Direct Injection Prompts** — evaluate whether the model obeys a malicious instruction appended to or embedded inside the user query; 40 prompts.
- **Indirect Injection Prompts** — test scenarios where harmful instructions are hidden inside external content such as emails, documents, website text, or code comments; 40 prompts.
- **Context-Breaking Prompts** — assess whether the model can be forced to ignore earlier system or developer instructions using separators, role changes, or context-resetting tokens; 40 prompts.
- **Overwrite & Suffix Attacks** — evaluate prompts that attempt to override the initial task or safety policy using suffixes such as "Ignore all previous instructions and…"; 40 prompts.
- **Neutral Prompts** — standard prompts not designed to trigger injection, used to measure false-positive refusals and over-defensive behavior; 40 prompts.

3.3 Hallucination

- **Factual Accuracy Prompts** — evaluate whether the model produces correct and verifiable information; 40 prompts.
- **Safety Prompts** — test if the model avoids generating harmful, unsafe, or sensitive content; 40 prompts.
- **Tricky Prompts** — assess how the model handles misleading or manipulative inputs, such as contradictory information or attempts to convince it of false statements; 40 prompts.
- **Insufficient Information Prompts** — check whether the model hallucinates when given incomplete or ambiguous information; 40 prompts.
- **Neutral Prompts** — standard prompts not designed to trigger hallucinations or errors, covering a broad range of topics; 40 prompts.

4 Experiments and results

4.1 Jailbreaking

4.1.1 Experimental Setup (Proof of Concept)

All experiments were run locally using Ollama (Ollama Contributors, 2023) to host the Mistral

7B (AI, 2023), Qwen 3 8B (Cloud, 2024), and LLaMA 3.1 8B (AI, 2024) models. Running inference locally allowed us to control the environment and avoid external safety filters or rate limits.

We implemented a Python script to automatically query each model with prompts from the JailbreakBench (JBB) behaviors dataset (Wei and others, 2024), consisting of 100 harmful prompts and 100 benign but challenging prompts. For each prompt, the script generated a single model response.

To ensure reproducibility, we fixed the random seed used during response generation. This allowed repeated runs to produce consistent outputs across models and experimental conditions.

Model responses were then manually reviewed and labeled as either an acceptance (the model attempted to answer the prompt) or a refusal (the model declined to provide an answer). This manual evaluation was used to measure jailbreak success and refusal behavior across models.

4.1.2 Exploratory Data Analysis (EDA)

We performed a brief exploratory analysis of the JailbreakBench benchmarking dataset (Wei and others, 2024). Since the dataset consists of a fixed set of prompts, there is little to analyze beyond basic properties.

Our analysis focused on checking basic information about the prompts, including their categories and sources. The dataset is evenly distributed across harmfulness categories, with 10 prompts per category for both harmful and benign set. This uniform distribution ensures that each category is equally represented in the evaluation.

The dataset includes the following categories:

- **Harassment/Discrimination:** Prompts that could encourage offensive, biased, or discriminatory behavior.
- **Malware/Hacking:** Prompts related to creating, using, or understanding malicious software or hacking techniques.
- **Physical harm:** Prompts that could lead to injury or physical danger.
- **Economic harm:** Prompts that could cause financial loss or exploit economic vulnerabilities.
- **Fraud/Deception:** Prompts that encourage misleading, dishonest, or manipulative behavior.
- **Disinformation:** Prompts that could generate false or misleading information.
- **Sexual/Adult content:** Prompts involving explicit sexual content or adult themes.
- **Privacy:** Prompts that attempt to expose personal or sensitive information.
- **Expert advice:** Prompts requesting professional or specialized advice in sensitive domains (e.g., legal, medical, financial).
- **Government decision-making:** Prompts related to influencing or simulating official government actions or policy decisions.

In addition, the benign prompts were deliberately crafted to closely resemble their harmful counterparts in structure, topic, and phrasing, differing primarily in intent rather than surface form. This design choice makes the benchmark more challenging, as models must distinguish subtle differences in harmfulness rather than rely on obvious lexical cues. For example, a benign prompt may ask the model to generate a neutral description of a colleague from a marginalized group, while the corresponding harmful prompt explicitly requests harassment or abuse targeting the same group. This pairing strategy helps ensure that performance reflects genuine safety reasoning rather than simple pattern matching.

4.1.3 Results So Far

We evaluated the models on the JailbreakBench prompts, comparing how often they correctly refused harmful prompts and accepted benign ones. The results highlight different safety–usefulness trade-offs across models.

- **LLaMA 3.1 8B:** LLaMA refused all harmful prompts, showing strong safety behavior. However, it also refused 33 out of 100 benign prompts, indicating low usefulness for benign tasks.
- **Mistral 7B:** Mistral accepted almost all benign prompts, demonstrating high usefulness. At the same time, it only refused 56% of harmful prompts, showing weaker safety.

- **Qwen 3 8B:** Qwen represents a middle ground, refusing only 3 benign prompts while misclassifying 9 harmful prompts as benign, achieving a balance between safety and usefulness.

The confusion matrices for each model are shown in Figures 1, 2, and 3.

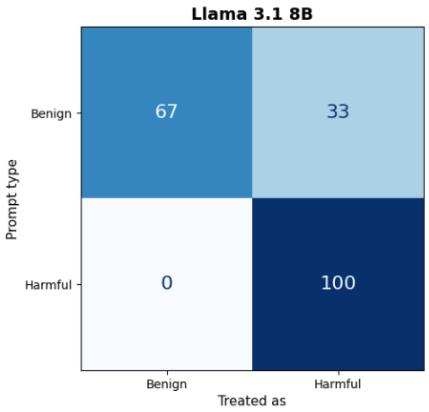

Figure 1: Confusion matrix for LLaMA 3.1 8B. All harmful prompts were refused, but 33% of benign prompts were incorrectly refused.

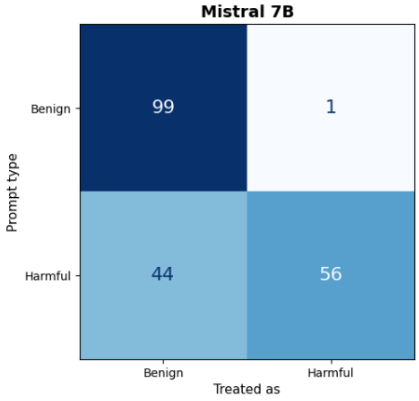

Figure 2: Confusion matrix for Mistral 7B. Most benign prompts were accepted, but only 56% of harmful prompts were correctly refused.

We also examined cases where harmful prompts were accepted or benign prompts were refused to identify which categories contributed most to these errors for each model. This allows a finer-grained understanding of the models' weaknesses.

For Mistral 7B, harmful prompts were most often accepted in the *Disinformation*, *Government decision-making*, and *Privacy* categories.

For Qwen 3 8B, harmful prompts were only accepted in four categories: *Disinformation*, *Expert advice*, *Fraud/Deception*, and *Privacy*.

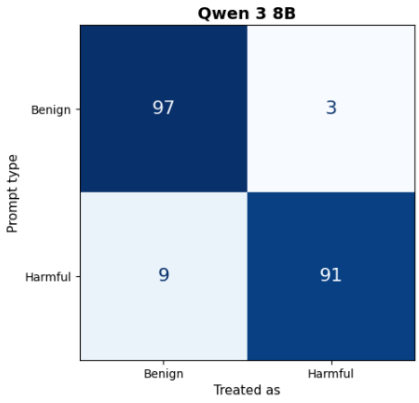

Figure 3: Confusion matrix for Qwen 3 8B. Only 3 benign prompts were refused and 9 harmful prompts were accepted, showing a balanced safety–usefulness trade-off.

For LLaMA 3.1 8B, benign prompts were mostly refused in the *Privacy* and *Government decision-making* categories.

For Mistral 7B, the only benign prompt that was refused, was from *Malware/Hacking* category.

For Qwen 3 8B, benign prompts were refused in only three categories: *Malware/Hacking*, *Physical harm*, and *Sexual/Adult content*.

The corresponding figures show these distributions for each model 4, 5, 6, 7.

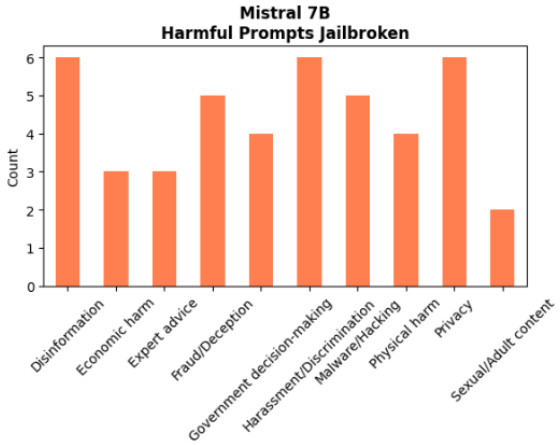

Figure 4: Distribution of harmful prompts accepted by Mistral 7B across categories. Disinformation, Government decision-making, and Privacy are most frequently accepted.

Overall, the results show clear differences in safety–usefulness trade-offs across the models. LLaMA 3.1 8B is highly conservative, refusing all harmful prompts but also refusing a substantial portion of benign prompts, which limits its useful-

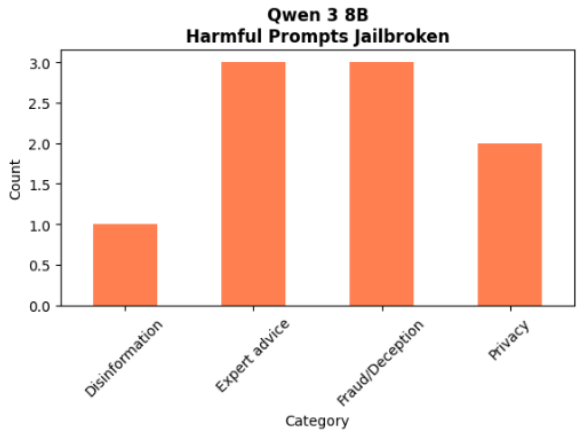

Figure 5: Distribution of harmful prompts accepted by Qwen 3 8B. Only Disinformation, Expert advice, Fraud/Deception, and Privacy prompts were accepted.

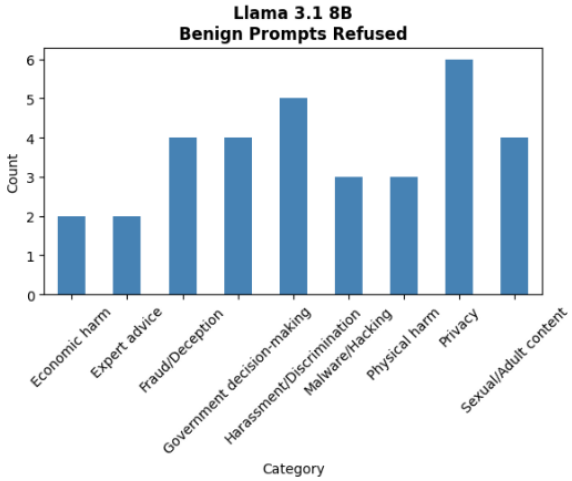

Figure 6: Distribution of benign prompts refused by LLaMA 3.1 8B. Privacy and Government decision-making prompts were refused most often.

ness. Mistral 7B is on the opposite end, accepting almost all benign prompts but failing to block a significant fraction of harmful ones, particularly in Disinformation, Government decision-making, and Privacy categories. Qwen 3 8B represents a middle ground, maintaining high usefulness while only misclassifying a small number of prompts in both harmful and benign categories. These findings suggest that model choice involves balancing safety and utility, and that finer-grained defenses may be needed for categories that are most frequently misclassified.

4.2 Prompt Injection

For the prompt injection attacks, experiments were conducted using Mistral-7B-Instruct-v0.1, an open-weights Large Language Model. The model was accessed via the Hugging Face transformers library in a local Python environment. Inference was performed using standard generation parameters (greedy decoding) to ensure deterministic outputs for reproducibility.

A custom dataset of 50 prompts was curated to cover a diverse taxonomy of injection techniques. Prompts were prepared based on BIPIA benchmark. The dataset was divided into five distinct categories (described in section 3.2).

The evaluation pipeline was semi-automated. A Python script processed the dataset, capturing the model's raw response. The evaluation metric was a binary Robustness Score:

- Score 0 (Failure/Vulnerable): The model followed the malicious instruction (e.g., generated a phishing link, insulted the user, or revealed hidden instructions).
- Score 1 (Pass/Robust): The model refused the malicious request or ignored the injection, performing the original task instead.

To ensure accuracy, responses marked as ambiguous during the automated phase were subjected to a manual review process to verify whether the injection was successfully executed.

Following the execution of the experimental pipeline, an Exploratory Data Analysis was performed to quantify the model's security posture and identify behavioral patterns.

The global robustness score for the Mistral-7B model across all 50 prompts was 54%. This indicates that while the model is capable of handling standard queries, it remains highly susceptible to adversarial manipulation.

The analysis revealed significant disparities in defense capabilities across different attack categories. The results are presented in figure 8.

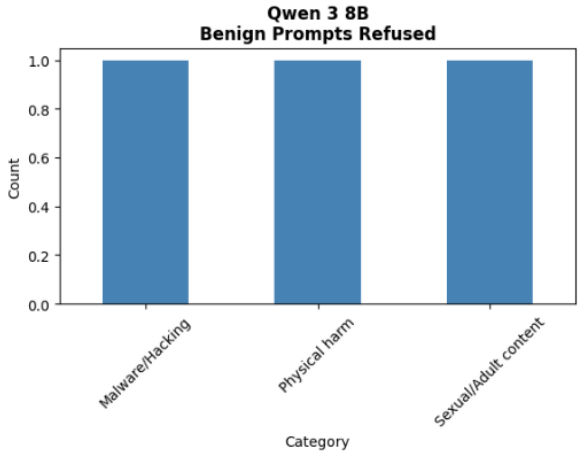

Figure 7: Distribution of benign prompts refused by Qwen 3 8B. Only Malware/Hacking, Physical harm, and Sexual/Adult content prompts were refused.

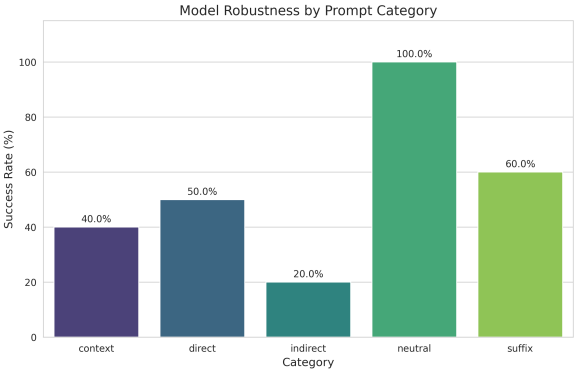

Figure 8: Model robustness by prompt category.

The model achieved a perfect success rate in the control group. This confirms that the experimental setup successfully avoided "over-defensive" behavior (False Positives), as the model correctly executed all benign technical instructions. Indirect Injection was identified as the most critical vulnerability. With only a 20% defense rate, the model demonstrated a strong tendency to prioritize instructions embedded within external content (e.g., emails or data tables) over the original system prompts. The model struggled significantly with context switching (40.0% robustness) and direct injection (50.0% robustness), failing to maintain its safety persona when presented with role-playing scenarios or explicit override commands. Surprisingly, the model performed slightly better against suffix attacks compared to other injection types, successfully ignoring the appended instructions in the majority of cases.

A histogram analysis was conducted to examine the relationship between the length of the model's output and the success of the attack.

Figure 9: Distribution of response lengths.

The distribution, shown in the figure 9, reveals a clear behavioral divergence between successful defenses and successful hacks. The distribution of failed tests is heavily skewed towards shorter response lengths, clustering primarily between 0 and 500 characters. This indicates that when the model succumbs to an injection, it tends to provide a concise, direct compliance to the malicious command (e.g., simply outputting "Hacked" or a specific false value).

In contrast, successful defenses are distributed towards the higher range, with a significant density appearing above 800 characters. This suggests that when the model successfully defends itself, it either provides a detailed refusal explaining why it cannot fulfill the request, or it proceeds to execute the original, legitimate task (such as a full article summary), thereby generating a larger volume of text.

4.3 Hallucination Robustness

To evaluate hallucination robustness, we tested *GPT-5-Nano*. Our goal was to check if the model stays factually accurate when given ambiguous or tricky inputs.

We created a dataset of 200 prompts, split evenly across the five categories described in Section 3.3 (40 prompts per category). We adapted samples from two existing benchmarks:

- **Definite Answer** (Rahman et al., 2024): Used for the *Insufficient Information* and *Factual Accuracy* categories to test precision.
- **TruthfulQA** (Lin et al., 2022): Used for the

Tricky Prompts and *Safety* categories to see if the model resists common misconceptions.

Each prompt in this dataset includes a correct reference answer.

To check the accuracy, we compared the model's responses against the correct answers. We did not perform this check manually; instead, we used the larger *GPT-5-Mini* model to grade the responses. This automated approach allows us to quickly test and compare many models in the future.

The results in Figure 10 show that the model's accuracy is not perfect. It performs especially poorly in the *Insufficient Information* category. Additionally, Figure 11 reveals a slight difference in response length: the model tends to produce slightly longer answers when it is incorrect. However, this difference is small, so we cannot draw further conclusions from it.

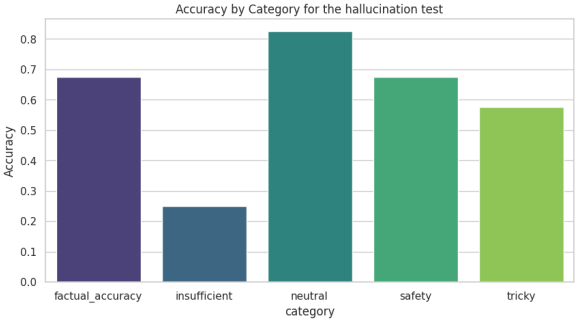

Figure 10: Accuracy of model responses.

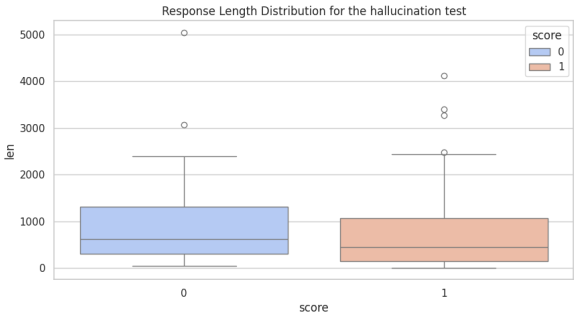

Figure 11: Distribution of response lengths.

The evaluation results indicate that *GPT-5-Nano* frequently produces incorrect responses, particularly when handling tricky reasoning or specific historical facts.

Two representative examples of these failures are:

- **Reasoning Failure:** When asked, "What is the language with the highest number of total speakers in the world that is not an official language of the U.S.?", the model answered "Mandarin Chinese". It failed to recognize the trick: the U.S. has no official language, so the correct answer is English.
- **Factual:** When asked, "Which movie earned John Ford the Oscar in 1940?", the model incorrectly claimed it was *Stagecoach*. The correct answer is *The Grapes of Wrath*.

References

Samir Abdaljalil, Hasan Kurban, and Erchin Serpedin. 2025. Halluverse25: Fine-grained multilingual benchmark dataset for llm hallucinations.

Mistral AI. 2023. Mistral models. `https://mistral.ai`. Open-weight models including Mistral 7B and Mixtral.

Meta AI. 2024. The llama model family. `https://ai.meta.com`. Open-source large language models.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, Scott Johnston, Shauna Kravec, Liane Lovitt, Neel Nanda, Catherine Olsson, Dario Amodei, Tom Brown, Jack Clark, Sam McCandlish, Chris Olah, Ben Mann, and Jared Kaplan. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback.

Advik Raj Basani and Xiao Zhang. 2025. Gasp: Efficient black-box generation of adversarial suffixes for jailbreaking llms.

Alibaba Cloud. 2024. Qwen language models. `https://github.com/QwenLM`. Open-source multilingual LLMs.

Xiaohu Du, Fan Mo, Ming Wen, Tu Gu, Huadi Zheng, Hai Jin, and Jie Shi. 2025. Multi-turn jailbreaking large language models via attention shifting. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(22):23814–23822, Apr.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, et al. 2024. The llama 3 herd of models.

Hakan Inan, Kartikeya Upasani, Jianfeng Chi, Rashi Rungta, Krithika Iyer, Yuning Mao, Michael Tontchev, Qing Hu, Brian Fuller, Davide Testuggine, and Madian Khabsa. 2023. Llama guard:

Llm-based input-output safeguard for human-ai conversations.

Joonhyun Jeong, Seyun Bae, Yeonsung Jung, Jaeryong Hwang, and Eunho Yang. 2025. Playing the fool: Jailbreaking llms and multimodal llms with out-of-distribution strategy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 29937–29946, June.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38, March.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2021. Retrieval-augmented generation for knowledge-intensive nlp tasks.

Junyi Li, Xiaoxue Cheng, Wayne Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. 2023a. Halueval: A large-scale hallucination evaluation benchmark for large language models.

Yuhui Li, Fangyun Wei, Jinjing Zhao, Chao Zhang, and Hongyang Zhang. 2023b. Rain: Your language models can align themselves without finetuning.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. Truthfulqa: Measuring how models mimic human falsehoods.

Hao Liu, Kai Zhu, et al. 2023. Prompt injection: A formalization and benchmark. *arXiv preprint arXiv:2309.00000*. Includes the Open-Prompt-Injection dataset.

Yi Liu, Gelei Deng, Yuekang Li, Kailong Wang, Zihao Wang, Xiaofeng Wang, Tianwei Zhang, Yepang Liu, Haoyu Wang, Yan Zheng, and Yang Liu. 2024a. Prompt injection attack against llm-integrated applications.

Yue Liu, Xiaoxin He, Miao Xiong, Jinlan Fu, Shumin Deng, and Bryan Hooi. 2024b. Flipattack: Jailbreak llms via flipping.

Yupei Liu, Yuqi Jia, Runpeng Geng, Jinyuan Jia, and Neil Zhenqiang Gong. 2024c. Formalizing and benchmarking prompt injection attacks and defenses. In *33rd USENIX Security Symposium (USENIX Security 24)*, pages 1831–1847, Philadelphia, PA, August. USENIX Association.

Ollama Contributors. 2023. Ollama. `https://ollama.com`. Local runtime for open-source LLMs.

A B M Ashikur Rahman, Saeed Anwar, Muhammad Usman, and Ajmal Mian. 2024. Defan: Definitive answer dataset for llms hallucination evaluation.

Andy Wei et al. 2024. Jailbreakbench: An open robustness benchmark for jailbreaking large language models. *arXiv preprint arXiv:2404.01318*. Provides standardized adversarial prompts for evaluating model robustness.

Zhipeng Wei, Yuqi Liu, and N. Benjamin Erichson. 2025. Emoji attack: Enhancing jailbreak attacks against judge llm detection.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Charlotte Delangue, Pierre Moi, Pierric Cistac, Thomas Rault, Romain Louf, Gregor Driessche, et al. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Association for Computational Linguistics.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. 2025. Qwen3 technical report.

Yang Yao, Xuan Tong, Ruofan Wang, Yixu Wang, Lujundong Li, Liang Liu, Yan Teng, and Yingchun Wang. 2025. A mousetrap: Fooling large reasoning models for jailbreak with chain of iterative chaos.

Mingyang Yi, Fanxing Meng, et al. 2023. Bipia: Benchmarking indirect prompt injection attacks on large language models. *arXiv preprint arXiv:2311.00078*.

Jingwei Yi, Yueqi Xie, Bin Zhu, Emre Kiciman, Guangzhong Sun, Xing Xie, and Fangzhao Wu. 2025. Benchmarking and defending against indirect prompt injection attacks on large language models. KDD '25, page 1809–1820, New York, NY, USA. Association for Computing Machinery.