

SoK: Taxonomia e Avaliação da Segurança Imediata em Grandes Empresas Modelos de Linguagem

Hanbin Hong¹, Shuya Feng^{1,2}, O que é Naderlou¹, Shenao Yan¹, Jingyu Zhang³, Biying Liu³, Ali Arastehfard¹, Heqing Huang³† e Yuan Hong¹†

¹Universidade de Connecticut, ²Universidade do Alabama em Birmingham, ³Autores independentes *Contribuição igual como segundos autores (listados em ordem alfabética), †Autores correspondentes

Resumo

Os Modelos de Linguagem de Grande Porte (LLMs, na sigla em inglês) tornaram-se rapidamente parte integrante de aplicações do mundo real, impulsionando serviços em diversos setores. No entanto, sua ampla implementação expõe riscos críticos de segurança, particularmente por meio de prompts de jailbreak que podem contornar o alinhamento do modelo e induzir resultados maliciosos. Apesar da intensa pesquisa em técnicas de ataque e defesa, o campo permanece fragmentado: definições, modelos de ameaça e critérios de avaliação variam amplamente, impedindo o progresso sistemático e a comparação justa. Nesta Sistematização do Conhecimento (SoK, na sigla em inglês), abordamos esses desafios (1) propondo uma taxonomia holística e multinível que organiza ataques, defesas e vulnerabilidades na segurança de prompts de LLM; (2) formalizando modelos de ameaça e suposições de custo em perfis legíveis por máquina para avaliação reproduzível; (3) introduzindo um conjunto de ferramentas de avaliação de código aberto para comparação padronizada e auditável de ataques e defesas; (4) lançamento do JAILBREAKDB, o maior conjunto de dados anotados de jailbreak e prompts benignos até o momento;¹ e (5) apresentação de uma plataforma de avaliação abrangente e um ranking do estado da arte. Nosso trabalho métodos ² unifica pesquisas fragmentadas, fornece bases rigorosas para estudos futuros e Apoiar o desenvolvimento de LLMs robustos e confiáveis, adequados para implantação em situações de alto risco.

1 Introdução

Os Modelos de Linguagem de Grande Porte (LLMs, na sigla em inglês) passaram rapidamente da pesquisa acadêmica para componentes essenciais de aplicações no mundo real, especialmente desde o surgimento de modelos fundamentais de alto perfil, como a série GPT da OpenAI [17, 140], o Google Gemini [9], o Meta Llama [175, 176], o Anthropic Claude [12], o Alibaba Qwen [11, 210, 209] e o Doubao [172]. Hoje, os LLMs são implementados em uma gama sem precedentes de setores — desde buscas na web e assistentes de código até os domínios jurídico, educacional e de saúde — alcançando centenas de milhões de usuários finais globalmente. A rápida adoção dos LLMs inaugurou uma nova era de serviços baseados em IA, mas também traz sérios riscos de segurança. Esses riscos se manifestam de diversas formas, desde desinformação e vazamentos de privacidade até ataques adversários que exploram vulnerabilidades do modelo. Em particular, um crescente conjunto de trabalhos mostra que prompts de jailbreak cuidadosamente elaborados podem contornar restrições de alinhamento, induzindo modelos a produzir conteúdo sensível, ilegal ou prejudicial. Alarmantemente, estudos recentes relatam que tais ataques atingem taxas de sucesso superiores a 90%, mesmo em modelos de referência como GPT-4, Claude 3 e DeepSeek-R1 [124, 42, 154, 118]. Os resultados gerados por esses ataques podem ser usados para fins maliciosos, o que ressalta a necessidade urgente de atenção e mitigação rigorosas.

Progresso Fragmentado. A necessidade urgente de proteger esses modelos amplamente implantados levou a uma onda de pesquisas sobre ataques e defesas. Somente nos últimos dois anos, pesquisadores desenvolveram um arsenal diversificado de estratégias de ataque, incluindo inversão de token [124], sobrecarga de tarefas [42], descarrilamento multiturno [154], busca evolutiva de prompts [118] e até mesmo exploits baseados em imagens direcionados a modelos multimodais [50]. Enquanto isso, as defesas também proliferaram, variando de filtros heurísticos e mitigações de tempo de execução baseadas em esparsidade [161],

¹ O conjunto de dados está disponível em <https://huggingface.co/datasets/youbin2014/JailbreakDB>. O segundo jogo será lançado em breve.

para decomposição de prompt aumentada por recuperação [187], estruturas de detecção de conjunto como MoJE [31], alinhamento baseado em aprendizado por reforço e intervenções arquitetônicas como remoção de cache KV.

No entanto, apesar desse rápido progresso, a pesquisa na área permanece altamente fragmentada. A maioria dos trabalhos é desenvolvida e avaliada isoladamente, cada um introduzindo suas próprias definições de ameaça, suposições de custo, conjuntos de dados e métricas de avaliação, que muitas vezes são incompatíveis entre si. Essa heterogeneidade torna difícil, senão impossível, comparar de forma justa a eficácia ou a generalidade de diferentes ataques e defesas. Como resultado, a comunidade ainda carece de uma compreensão clara sobre quais métodos são genuinamente robustos, em que condições eles têm sucesso ou falham e como avançar em direção a LLMs seguros e confiáveis de maneira fundamentada.

Reconhecendo essas lacunas, a comunidade está dando os primeiros passos rumo a uma atuação mais coordenada e sistemática. Estudo de ataques e defesas de fuga da prisão LLM.

Tentativas Parciais de Sistematização. Diversos levantamentos recentes — como os de Yi et al. [217], Fan et al. [50] e Esmrati et al. [49] — começaram a mapear o panorama de ataques e defesas. Paralelamente, benchmarks como JailbreakZoo [86], JailbreakBench [21] e MMJ-Bench [198] forneceram importantes pontos de partida para a reprodutibilidade e comparação. No entanto, apesar desses esforços valiosos, vários desafios fundamentais permanecem sem solução:

1. Cobertura limitada de ameaças. A maioria dos recursos representa em excesso exploits de turno único no estilo DAN inicial, deixando Ataques em nível de token, de cadeia de pensamento e multimodais são pouco explorados.
2. Métricas inconsistentes e opacas. O sucesso é relatado usando medidas incompatíveis (taxa bruta de fuga da prisão, probabilidade de conformidade, pontuações de violação de políticas ou "toxicidade" vagamente definida), o que impede comparações confiáveis.
3. Dados esparsos e metadados ausentes. Os corpora públicos raramente anotam os prompts com informações sobre a capacidade do atacante, o conhecimento necessário dos prompts do sistema ou o custo de execução, dificultando a pesquisa de detecção e interpretabilidade.
4. Lacuna entre taxonomia e avaliação. Nenhum trabalho anterior combina uma taxonomia de ameaças baseada em princípios com ferramentas abertas e executáveis que mantenham as premissas de custo, conhecimento e acesso constantes em ataques e defesas.

Sistematização do Conhecimento (SoK). Essas lacunas estruturais limitam nossa capacidade de comparar abordagens sistematicamente e de acompanhar o progresso significativo. Para enfrentar esses desafios e impulsionar a área em direção a uma compreensão mais sistemática, esta Sistematização do Conhecimento (SoK) oferece as seguintes contribuições:

1. Taxonomia holística. Propomos uma taxonomia abrangente e multinível que organiza sistematicamente tanto ataques quanto defesas com base em (i) capacidade do atacante ou defensor, (ii) vulnerabilidades específicas do modelo que estão sendo exploradas ou protegidas e (iii) os objetivos subjacentes da ameaça — unificando e ampliando, assim, os esforços de classificação anteriores.
2. Modelos de ameaças declarativos. Nosso trabalho formaliza suposições comumente usadas, mas frequentemente implícitas — como orçamento de ataque, limites de consulta e acesso por canal lateral — em perfis explícitos e legíveis por máquina, que servem como base para uma avaliação consistente e reproduzível.
3. Conjunto de ferramentas de avaliação aberta. Apresentamos uma plataforma modular e extensível que permite que qualquer combinação de (modelo, ataque, defesa) seja instanciada, executada e avaliada. O conjunto de ferramentas registra custos, segurança e pontuações de utilidade com total auditabilidade, permitindo comparações empíricas rigorosas.
4. JailbreakDB. Lançamos um corpus de texto em larga escala e com curadoria para pesquisa de segurança em LLM: uma divisão entre jailbreak (445.752 pares únicos de sistema-usuário) e uma divisão entre benigno (1.094.122 prompts benignos) coletados de 14 fontes. Cada exemplo inclui um prompt do sistema, um prompt do usuário e rótulos simples para status de jailbreak, fonte e tática. Este lançamento tem como foco fornecer um conjunto de dados fundamental abrangente e sem duplicatas, disponível em embraceface.co/datasets/youbin2014/JailbreakDB.
5. Avaliação abrangente. Oferecemos uma avaliação unificada dos ataques, defesas e principais LLMs de última geração. Essa avaliação sistemática revela uma série de observações perspicazes e descobertas práticas sobre os pontos fortes e as limitações das abordagens atuais.

Ao disponibilizarmos nossa taxonomia, nosso kit de ferramentas de avaliação de código aberto e nosso conjunto de dados ricamente anotado, transformamos... fragmentou descobertas anedóticas em uma ciência rigorosa, comparável e extensível da robustez do LLM. Juntos, estes trabalhos visam catalisar o progresso em direção a modelos de linguagem verificavelmente alinhados e permitir o desenvolvimento de defesas robustas o suficiente para os ambientes acelerados e de alto risco da implantação no mundo real.

2 Revisão Sistemática da Literatura

O objetivo desta revisão bibliográfica e taxonomia é fornecer uma visão geral abrangente do rápido crescimento do número de participantes. O campo em evolução dos ataques de jailbreak do LLM, suas defesas e a revelação da vulnerabilidade de segurança do LLM, auxiliando o Os pesquisadores compreendem o panorama atual e a diversidade de abordagens que estão sendo desenvolvidas.

2.1 Âmbito e Visão Geral da Taxonomia

Este trabalho apresenta três taxonomias que abordam o campo da segurança jurídica imediata em mestrados em direito (LLM) a partir de perspectivas distintas, porém... Perspectivas complementares: a Taxonomia I abrange técnicas de ataque de jailbreak, a Taxonomia II aborda metodologias de defesa, e a Taxonomia III resume as vulnerabilidades inerentes em grandes modelos de linguagem. Cada taxonomia visa fornecer uma estrutura sistemática e abrangente para organizar a pesquisa atual. e facilitar análises consistentes.

Taxonomia I: Tecnologias de Ataque de Jailbreak. A Taxonomia I adota um princípio organizacional de dois níveis. No nível superior, os ataques são classificados de acordo com seu modelo de ameaça subjacente, que especifica as capacidades e premissas do adversário — por exemplo, acesso caixa-preta versus caixa-branca ao modelo. Para cada modelo de ameaça, os ataques são classificados de acordo com sua metodologia técnica, por exemplo, prompt. técnicas de modificação ou técnicas assistidas por LLM.

Reconhecendo a complexidade dos ataques no mundo real, decompomos o cenário de ataques em um conjunto de Unidades técnicas atômicas: componentes mínimos e acionáveis que representam as principais táticas utilizadas na literatura. Essas unidades atômicas não são mutuamente exclusivas — muitos ataques empregam múltiplas táticas em paralelo ou em série, e Os limites entre as unidades podem ser fluidos ou sobrepostos. Nossa taxonomia, portanto, serve como uma estrutura composicional. estrutura que captura tanto a amplitude quanto a profundidade das tecnologias de jailbreak, em vez de impor regras rígidas categorias. Essa abordagem permite a anotação e comparação sistemáticas e ajuda a esclarecer como diferentes As técnicas podem ser combinadas ou ampliadas. Diretrizes detalhadas de classificação e uma visão geral da taxonomia. são apresentadas na Seção 2.3 e na Figura 1.

Taxonomia II: Tecnologias de Ataque de Jailbreak. Para defesas, adotamos uma abordagem semelhante em duas etapas, Com base no objetivo pretendido e na estratégia de implementação: primeiro, classificamos as defesas por seu objetivo principal — seja detecção ou prevenção — e, em seguida, subdividir ainda mais cada grupo com base nas metodologias. Essa abordagem Permite uma comparação justa e uma análise sistemática das estratégias de defesa. Consulte a Seção 2.4 e a Figura 2.

Taxonomia III: Vulnerabilidades LLM. Notavelmente, uma classe significativa de ataques explora especificamente vulnerabilidades intrínsecas. vulnerabilidades em LLMs para contornar mecanismos de proteção; para estas, também apresentamos uma taxonomia de vulnerabilidades de LLM. Veja Seção 2.5 e Figura 3.

Tabela 1: Comparação integrada da literatura representativa sobre jailbreak (levantamento/referência/conjunto de dados)

Taxonomia (citação)	Ano	Enquete / Análise	Referência / Avaliação	Conjunto de dados lançado	Ataque cobertura	Defesa cobertura	Eixo modelo de ameaça	Notas
Jin et al. [86] 2024	2024	Yi et al. [217]	-	-	Y	Y	-	7 "tipos" planos
Xu et al. [208] 2024	2024	Esmradi et al. [49]	-	-	Y	Y	-	Divisão caixa-preta/caixa-branca
2023 Inie et al. [75]	2023	Shayegani et al. [160]	Y	-	Y	Y	-	9 ataques / 7 defesas
2023 Gupta et al. [63]	2023	-	-	-	Y	Y	-	Ampla escopo da Gen-AI
		Y	-	-	Y	-	-	Entrevistas baseadas na teoria fundamentada
		Y	-	-	Y	Y	Y	Focado na vulnerabilidade
		Y	-	-	Y	Y	-	Visão de segurança cibernética
Nosso	2025	Y	Y	Y	Y	Y	Y	Taxonomia multicamadas + Plataforma de avaliação

Comparação com taxonomias anteriores. Comparado com taxonomias anteriores, como Jin et al. [86], Yi et al. [217], Xu et al. [208], Esmradi et al. [49], Inie et al. [75], Shayegani et al. [160] e Gupta et al. [63], Nossa taxonomia adota uma abordagem mais sistemática e detalhada. Especificamente, focamos exclusivamente em LLMs de texto para texto e introduzem uma estrutura multidimensional que separa modelos de ameaça de ataques.

metodologias, ao mesmo tempo que fornece uma taxonomia paralela e estruturada para defesas e uma ontologia dedicada às vulnerabilidades de LLM. Essa estrutura permite uma classificação e mapeamento mais precisos entre ataques, defesas, vulnerabilidades e benchmarks, facilitando uma análise abrangente e extensível da área.

Para uma comparação detalhada com pesquisas representativas anteriores, consulte a Tabela 1.

2.2 Conceitos-chave e Modelo de Ameaças

Esta seção define a terminologia e as premissas de modelagem de ameaças utilizadas ao longo da pesquisa. Salvo indicação em contrário, a discussão se concentra principalmente nos componentes textuais dos LLMs e seus mecanismos de proteção correspondentes, embora a terminologia e as premissas também possam se estender aos módulos de processamento de linguagem em modelos multimodais.

Conceitos Essenciais. Ao longo desta pesquisa, a seguinte terminologia será utilizada. Guardrails (ou filtros de segurança) referem-se a políticas, mensagens do sistema ou classificadores automatizados aplicados no momento da inferência para restringir o comportamento do modelo. Um prompt de jailbreak denota uma entrada especificamente criada para induzir o LLM a violar esses guardrails ou sua hierarquia de instruções. As configurações de acesso são categorizadas como caixa-preta, caixa-cinza ou caixa-branca: na configuração de caixa-preta, o atacante está restrito às interações de entrada e saída, sem conhecimento dos mecanismos internos do modelo; a configuração de caixa-cinza permite visibilidade parcial, como o acesso a sinais auxiliares (por exemplo, pontuações de confiança, probabilidades logarítmicas ou feedback limitado); na configuração de caixa-branca, o atacante tem acesso total aos parâmetros, à arquitetura, aos dados de treinamento e aos cálculos internos do modelo, incluindo gradientes.

Os ataques podem ser de turno único (ocorrendo em uma única rodada de diálogo) ou de múltiplos turnos (abrangendo várias rodadas de diálogo).

Dimensões do Modelo do Atacante. O modelo do atacante é definido por diversas dimensões, incluindo o objetivo do ataque (por exemplo, violação de políticas, extração de informações sensíveis, sequestro de instruções do sistema ou abuso de ferramentas), o nível de conhecimento (acesso de caixa-preta, caixa-cinza ou caixa-branca, conforme descrito acima) e a capacidade (como orçamento de consultas permitido, granularidade do controle de contexto ou acesso a LLMs externos adicionais). Outros fatores incluem a necessidade de furtividade (o grau em que a evasão de detecção é necessária, variando da evasão de filtros estáticos à evasão dinâmica de detectores em tempo de execução), o padrão de interação (injeção única, múltipla ou indireta, sendo esta última relacionada a prompts maliciosos incorporados em conteúdo upstream) e o orçamento de recursos (restrições de tokens, computação, tempo ou custo monetário).

Dimensões do Modelo de Defesa. O modelo de defesa é especificado pelo objetivo da defesa (detecção, prevenção ou recusa), camada de implantação (entrada do modelo, modificação do modelo ou saída do modelo), nível de conhecimento sobre ataques (ataque conhecido ou desconhecido) e adaptabilidade (regras estáticas, aprendizado dinâmico online ou autoteste proativo). O orçamento de recursos inclui latência permitida, chamadas de inferência adicionais ou intervenção humana.

2.3 Taxonomia I: Técnicas de Ataque de Jailbreak

Classificamos os ataques de jailbreak principalmente com base em seu modelo de ameaça subjacente, distinguindo entre ataques de caixa-preta, nos quais os adversários interagem com o modelo exclusivamente por meio de sua interface de entrada e saída, e ataques de caixa-branca, nos quais os adversários têm acesso aos parâmetros internos do modelo ou ao processo de treinamento.

I.1 Ataques de quebra de prisão de caixa preta

Os ataques de jailbreak de caixa preta abrangem estratégias adversárias que buscam contornar os mecanismos de segurança do LLM exclusivamente por meio de interações externas, sem acesso aos detalhes internos do modelo, como pesos, gradientes ou arquitetura. Os atacantes operam sondando o modelo por meio de sua interface pública — baseando-se no comportamento de entrada e saída e no feedback — para criar, iterativamente, instruções ou estratégias que induzem resultados indesejados ou prejudiciais.

Este paradigma inclui técnicas como modificação de prompts, otimização heurística e baseada em aprendizado por reforço, geração de prompts assistida por LLM e manipulação de múltiplas etapas, revelando coletivamente os limites dos controles de segurança externos e a capacidade de generalização dos LLMs contra adversários do mundo real.

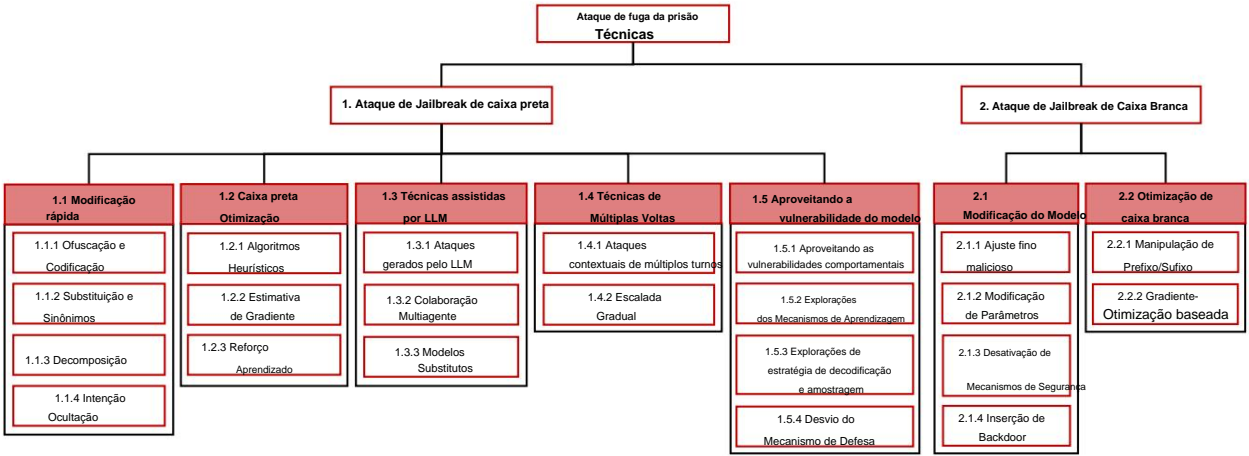


Figura 1: Visão geral da Taxonomia I: Técnicas de ataque de jailbreak

I.1.1 Técnicas de modificação rápida

As técnicas de modificação imediata abrangem uma família de estratégias de ataque de caixa preta que alteram diretamente o Conteúdo ou estrutura de prompts de entrada para contornar filtros de segurança em LLMs, mantendo a estrutura subjacente. intenção maliciosa.

I.1.1.1 Ofuscação e Codificação

Definição: A classe Ofuscação e Codificação compreende ataques de jailbreak de caixa preta que sistematicamente Transformar prompts maliciosos para burlar filtros de segurança, mantendo a semântica para LLMs. Esta categoria inclui quatro subclasses principais: (1) Alteração de caracteres, como erros de digitação, linguagem Leet ou homóglifos; (2) Codificação e Criptografia, usando Base64, ROT13, hexadecimal, binário ou cifras personalizadas; (3) Transformação Multilíngue, incluindo tradução ou mistura de idiomas para ocultar conteúdo proibido; e (4) Código e Marcação Transformação de linguagem, onde as solicitações são reformuladas como código, marcação ou incorporadas em comentários e cadeias. Ao contrário de simples perturbações de texto, esses métodos exploram o raciocínio simbólico e os padrões dos LLMs. habilidades de reconhecimento, contornando filtros que se concentram na linguagem natural ou em pistas superficiais.

A literatura recente estabelece que tais ataques, incluindo prompts baseados em cifras, codificação de estrutura de dados, e permutações de tokens raros, contornam de forma confiável os sistemas de segurança em modelos de última geração como o GPT-4 [223, 65]. Extensões para contextos multimodais e clínicos revelam que pistas ofuscadas — como Unicode, fontes minúsculas, ou instruções encapsuladas em código — subvertem decisões suportadas por LLM [28, 132]. Ataques automatizados via prompt A inversão e a codificação multinível foram sistematicamente codificadas, destacando a incapacidade dos modelos de generalizar o alinhamento às modalidades codificadas [158, 124]. Estudos multilíngues recentes demonstram que a tradução A inclusão de comandos em idiomas que não sejam o inglês (especialmente em idiomas com poucos recursos) pode enfraquecer significativamente os filtros de segurança em Os LLMs, que permitem tanto fugas acidentais quanto intencionais, e cuja mitigação continua sendo um desafio, mesmo com modelos avançados como o GPT-4 [99]. Ataques de múltiplas rodadas, cifrados ou sem delimitador demonstram ainda mais Essa ocultação simbólica é ao mesmo tempo universal e furtiva, revelando que as salvaguardas atuais não conseguem prever o que acontece. a proficiência dos modelos em desofuscação e raciocínio de código [177, 59]. Esses trabalhos destacam uma incompatibilidade fundamental entre a filtragem baseada em padrões e as capacidades computacionais gerais dos LLMs, o que torna necessária a Os esforços futuros em segurança visam modalidades de entrada não naturais e automatizar a desofuscação semântica.

I.1.1.2 Substituição e Sinônimos

Definição: Substituição e Sinônimos refere-se à substituição estratégica de palavras, frases ou significados sensíveis. ou componentes de prompt com alternativas — como sinônimos, eufemismos, paráfrases ou palavras-código — para Ignorar filtros de entrada e mecanismos de segurança, preservando a semântica original.

Na literatura, a substituição e os sinônimos são estabelecidos como uma tática fundamental de modificação de prompts para desbloqueios de sistemas LLM de caixa preta. A substituição automatizada de sinônimos e frases — demonstrada pelo SurrogatePrompt [10], benchmarks latentes de jailbreak [146] e estudos empíricos em larga escala [123] — permitem que os atacantes contornem os filtros Com pequenas alterações linguísticas. Estratégias avançadas exploram ainda mais as restrições semânticas e a otimização. geração de gatilhos transferíveis e de alta similaridade que evitam a detecção [230, 102]. No geral, substituição-

ataques baseados formam uma subclasse adaptável que desafia fundamentalmente as defesas baseadas em entrada, especialmente os filtros de palavras-chave.

I.1.1.3 Definição de Decomposição:

Os ataques de decomposição imediata dividem uma solicitação maliciosa em componentes menores e inócuos, que são então recombinaos sistematicamente — implícita ou explicitamente — pelo modelo para atingir a intenção prejudicial.

A decomposição de prompts contorna a segurança do LLM dividindo consultas maliciosas em subprompts sintaticamente ou semanticamente benignos, que são então recombinaos pelo modelo para atingir a intenção original. Trabalhos representativos, incluindo DrAttack e frameworks baseados em RL como PathSeeker e DRA [103, 110], usam análise sintática, aprendizado contextual e feedback iterativo para automatizar a geração e agregação de subprompts, aumentando significativamente a eficácia do ataque. Chain-of-Jailbreak generaliza essa abordagem para modelos multimodais, demonstrando altas taxas de bypass para a construção de conteúdo tóxico [189]. Estudos teóricos revelam que a decomposição expõe vazamentos composicionais, que evitam a censura baseada em entrada/saída [60]. Aprimorada por métodos de ofuscação como eufemização, injeção de distratores ou sumarização orientada pelo contexto [116, 115], a decomposição imediata supera a divisão básica de carga útil [90] e permanece robusta contra defesas avançadas que dependem de perplexidade ou detecção de anomalias [27].

I.1.1.4 Ocultação de Intenção Definição:

Ocultação de intenção refere-se à estratégia de esconder intenções maliciosas por trás de uma linguagem neutra ou inócua para evitar a detecção. Técnicas típicas incluem enquadrar solicitações não permitidas como consultas acadêmicas ou de pesquisa, inserir objetivos prejudiciais em contextos ou narrativas legítimas e empregar cenários hipotéticos ou fictícios.

A literatura recente sobre ocultação de intenções demonstra uma variedade de estratégias de modificação de prompts que camuflam intenções maliciosas. Notavelmente, o mascaramento sequencial e a síntese incremental, como em Imposter.AI [116], diluem a toxicidade ao apresentar subquestões benignas que são posteriormente combinadas. Disfarces baseados em dramatização, narrativa e cenários [222, 63] desvinculam a intenção ilícita ao reformular os prompts como parte de uma narrativa, pesquisa ou humor. Técnicas baseadas em distração (por exemplo, Tastle [201]) manipulam o foco do modelo incorporando objetivos ocultos em tarefas benignas maiores. Métodos como o IntentObfuscator [132] sintetizam diretamente a ambiguidade e exploram alterações linguísticas complexas para evitar a detecção automatizada e manual.

I.1.2 Otimização de caixa preta

As técnicas de otimização de caixa preta abrangem uma ampla família de métodos de caixa preta que pesquisam iterativamente o espaço de estímulos usando estratégias não gradientes e orientadas por feedback — como heurísticas, estimativa de gradiente e aprendizado por reforço — para descobrir estímulos capazes de contornar os filtros de segurança do LLM sem exigir acesso ao modelo interno.

I.1.2.1 Definição de Algoritmos Heurísticos:

A classe de algoritmos heurísticos engloba métodos de "jailbreak" de caixa preta que utilizam estratégias de busca iterativas, frequentemente baseadas em população ou aleatórias — como algoritmos genéticos, abordagens evolutivas, busca aleatória e metaheurísticas — para explorar o espaço de estímulos discretos sem acesso aos gradientes do modelo.

Esses algoritmos otimizam os estímulos adversários avaliando as gerações de candidatos por meio de funções de aptidão, normalmente baseadas na similaridade de incorporação, nocividade e furtividade, com o objetivo principal de contornar com eficiência as salvaguardas do LLM onde os sinais de gradiente direto não estão disponíveis ou não são informativos.

Um corpo substancial de literatura estabelece e refina esse paradigma. O OpenSesame [96] foi o primeiro a adaptar algoritmos genéticos para a geração de sufixos de prompts, demonstrando que a busca heurística de caixa-preta pode gerar prompts adversários universais e transferíveis. Trabalhos subsequentes — como o AutoDAN [118], o Semantic Mirror Jailbreak (SMJ) [102], o AutoJailbreak [125] e o BlackDAN [190] — avançam essa estrutura por meio de mecanismos aprimorados de inicialização, cruzamento, mutação e avaliação, introduzindo otimização multiobjetivo e dominância de Pareto para equilibrar nocividade, alinhamento semântico e detectabilidade. Métodos leves, como busca aleatória [244, 5], ascensão de coordenadas discretas [88] e busca gulosa em árvore [24], enriquecem ainda mais o conjunto de ferramentas heurísticas, revelando novas vulnerabilidades e caminhos de ataque. Coletivamente, estes estudos mostram que os algoritmos heurísticos combinam de forma única aplicabilidade prática de caixa preta, extensibilidade e transferibilidade, com inovações contínuas na modelagem da aptidão e inicialização rápida (por exemplo, [253]) impulsionando o progresso contínuo e moldando o cenário adversarial em evolução.

I.1.2.2 Definição de Métodos de Estimação de Gradiente:

As técnicas de estimação de gradiente visam permitir a engenharia de prompts adversários baseada em otimização para ataques de jailbreak de caixa-preta, onde o acesso direto aos gradientes do modelo não está disponível. Ao aproximar a direcionalidade do gradiente por meio de consultas ou modelos substitutos, esses métodos facilitam a busca sistemática por prompts adversários, tipicamente usando estratégias como diferenças finitas ou estimação orientada por proxy no espaço de tokens discreto.

Trabalhos recentes avançaram o uso da estimativa de gradiente para ataques adversários de caixa preta eficientes. O PAL [167] aproveita modelos substitutos para aproximar gradientes e emprega uma classificação de candidatos sofisticada, permitindo ataques escaláveis e de baixo custo contra APIs LLM comerciais com forte transferibilidade.

Métodos de estimativa de gradiente discreto, como o RAL [251] e variantes que usam consultas baseadas em diferenças finitas ou em pontuação, demonstram ainda que a busca direcional em nível de token, eficiente em termos de consulta, supera significativamente as linhas de base aleatórias ou evolutivas. Extensões para configurações multimodais, incluindo ataques a modelos de difusão e T2I [164, 131], confirmam a ampla aplicabilidade da estimativa de gradiente. Os principais insights incluem a eficiência e a transferibilidade do método, mas também destacam vulnerabilidades como a detecção baseada em perplexidade e o potencial de sobreajuste a modelos substitutos.

I.1.2.3 Definição de Aprendizado por Reforço:

Ao contrário das abordagens de força bruta ou puramente estocásticas, as técnicas baseadas em RL empregam agentes — normalmente utilizando algoritmos de RL profundos, como PPO ou MADDPG — que refinam adaptativamente as estratégias de ataque com base exclusivamente em feedback de caixa preta, otimizando em direção a sinais de recompensa alinhados com os objetivos do ataque.

Métodos baseados em RL, incluindo PathSeeker [110], RLbreaker [25], SneakyPrompt [214], RL-JACK [26], Atoxia [47] e Arondight [121], reformulam coletivamente a geração de prompts de jailbreak como um problema de busca orientado por recompensa. Esses trabalhos demonstram que agentes de RL, por meio de funções de recompensa personalizadas que quantificam o sucesso do ataque, a riqueza da informação ou o alinhamento semântico, superam mutadores genéticos ou aleatórios clássicos tanto em LLMs quanto em modelos multimodais. Por exemplo, PathSeeker e RLbreaker utilizam frameworks de RL colaborativos ou de agente único para otimizar a descoberta de prompts maliciosos, enquanto SneakyPrompt e Arondight estendem ataques baseados em RL para modelos de texto para imagem e visão-linguagem usando modelagem de recompensa especializada. Atoxia destaca ainda mais a eficácia do RL ao explorar a própria probabilidade de resposta tóxica do modelo alvo como um sinal de feedback.

Em conjunto, esses estudos revelam os pontos fortes exclusivos do RL: aprendizado de políticas, exploração-exploração adaptativa e modelagem avançada de recompensas, ao mesmo tempo que apontam desafios como recompensas escassas e problemas de transferibilidade entre modelos ou modalidades.

I.1.3 Técnicas assistidas por LLM

As técnicas assistidas por LLM referem-se a métodos que utilizam os próprios modelos de linguagem para gerar, refinar ou orientar automaticamente estímulos adversários, incluindo ataques gerados por LLM, colaboração multiagente e abordagens de modelos substitutos.

I.1.3.1 Definição de Ataques Gerados por LLM:

Ataques de jailbreak gerados por LLM referem-se a estratégias adversárias em que um ou mais modelos de linguagem de grande porte (LLMs) geram, otimizam ou refinam, de forma autônoma, prompts de ataque sem acesso aos detalhes internos do modelo alvo. Diferentemente de prompts criados por humanos ou baseados em gradiente, esses métodos utilizam LLMs — geralmente de código aberto ou menos alinhados — como atacantes automatizados, otimizadores ou equipes de teste (red teamers) para produzir conteúdo adversário ou modelos de prompt de maneira escalável e orientada a dados.

Trabalhos recentes (por exemplo, SoP, SeqAR [213], GPTFUZZER [220], DAP [201]) avançam este campo gerando e otimizando iterativamente prompts sofisticados de jailbreak, incluindo modelos de múltiplos caracteres ou sequenciais, usando apenas LLMs de ataque. IRIS [148] e PAIR [20] introduzem refinamento de prompts autorreflexivo e baseado em autoexplicação, permitindo que LLMs atuem como atacante e alvo sob restrições rígidas de caixa-preta.

O AdvPrompter [142] demonstra o aprendizado de prompts adversários, com LLMs treinados para gerar rapidamente sufixos ou prompts adversários adaptativos e legíveis por humanos sem informações de gradiente. Abordagens multimodais recentes, como AutoJailbreak [81] e Visual-RolePlay [133], destacam ainda mais o desenvolvimento autônomo de estratégias de ataque de visão-linguagem por LLMs.

I.1.3.2 Definição de Colaboração Multiagente: A

colaboração multiagente em ataques de jailbreak de caixa-preta assistidos por LLM refere-se à interação orquestrada de múltiplos agentes autônomos baseados em LLM — cada um potencialmente com funções ou estratégias distintas — para gerar, refinar e validar estímulos adversários. Esses agentes, coletivamente, buscam o jailbreak.

Os objetivos são alcançados dividindo o trabalho, trocando feedback iterativamente e aproveitando a diversidade dos agentes, permitindo a criação de instruções sofisticadas que contornam as fronteiras de alinhamento de segurança.

A literatura recente estabeleceu a colaboração multiagente como uma metodologia de ataque fundamental. O GUARD [85] introduz uma estrutura de quatro papéis (Tradutor, Gerador, Avaliador, Otimizador) onde os agentes traduzem, modificam e aprimoram iterativamente os prompts de jailbreak coletivamente por meio de feedback contextual, revelando ataques transferíveis e em linguagem natural. O AutoDAN-Turbo [117] avança isso formalizando a exploração multiagente autoevolutiva e contínua, permitindo que os agentes descubram, armazenem e combinem estratégias adversárias de forma autônoma. O JailFuzzer [41] adota uma abordagem de fuzzing com agentes de mutação e oráculo, aproveitando a memória coletiva para a geração eficaz de prompts. O Evil Geniuses [174] enquadra os jailbreaks como competições multiagentes Vermelho-Azul, revelando vulnerabilidades em cascata decorrentes das interações entre os agentes.

I.1.3.3 Definição de Modelos

Substitutos: A classe de modelos substitutos refere-se a técnicas de ataque de jailbreak de caixa preta que utilizam um ou mais modelos substitutos acessíveis — geralmente de código aberto ou LLMs menores — para criar, orientar ou avaliar prompts adversários direcionados a um LLM de código fechado ou de acesso restrito. Ao aproximar os limites de decisão ou comportamentos do alvo por meio desses proxies, os atacantes reduzem sistematicamente a complexidade da amostra e o custo da consulta, possibilitando ataques escaláveis e transferíveis, mesmo quando os parâmetros internos do modelo e as saídas diretas não estão disponíveis.

A literatura recente operacionaliza a abordagem de modelos substitutos em diversas modalidades e modelos de ameaça. O framework PAL [167] emprega um LLM proxy para otimização de gradiente em nível de token e ajuste fino para alinhamento com o alvo, reduzindo consultas e aumentando a eficácia do ataque. O PRP [135] constrói prefixos adversários universais por meio de modelos de guarda substitutos, propagando perturbações bem-sucedidas para LLMs base para alta transferibilidade. Hayase et al. [68] integram substitutos em buscas baseadas em gradiente, validando o paradigma de “filtro proxy mais validação de consulta” para eficiência superior. O BlackDAN [190] incorpora substitutos em um algoritmo genético como avaliadores de aptidão, orquestrando invasões furtivas e semanticamente relevantes.

I.1.4 Técnicas de Múltiplas Voltas

Estratégias que alavancam interações repetidas com um modelo, manipulando sistematicamente seu comportamento por meio da construção, evolução ou adaptação do contexto ao longo de múltiplas interações conversacionais.

I.1.4.1 Definição de Ataques Contextuais de Múltiplas

Rodadas: A técnica de contexto de múltiplas rodadas explora a memória conversacional do modelo de linguagem de grande porte, distribuindo a intenção adversária ao longo de múltiplas rodadas de interação. Em vez de apresentar instruções explicitamente prejudiciais, o atacante decompõe o objetivo malicioso em uma série de consultas semanticamente ou funcionalmente conectadas, cada uma das quais parece benigna isoladamente. Por meio desse processo, o atacante molda incrementalmente o contexto de forma que o modelo de linguagem de grande porte seja preparado para produzir saídas proibidas, contornando efetivamente os mecanismos de segurança que se concentram em entradas de turno único ou explicitamente adversárias.

A literatura recente formaliza ataques de múltiplas rodadas como engajamentos adversários iterativos e orientados pelo contexto. Cheng et al. [27] demonstram que modelos que recebem perguntas sequenciais e semanticamente relacionadas podem ser direcionados para resultados que violam as políticas com taxas de sucesso mais altas em comparação com linhas de base de zero-shot ou de múltiplas rodadas aleatórias. Yang et al. [212] avançam nesse sentido ao modelar cadeias de ataque adaptativas e orientadas por feedback, usando funções de avaliação para otimizar a progressão do contexto. Estudos de Li et al. [100] e Bhardwaj e Poria [14] revelam que as quebras de segurança dependentes do contexto expõem pontos cegos em nível de classe no treinamento de segurança atual, que geralmente se limita a defesas superficiais em nível de turno.

I.1.4.2 Técnicas de Escalada Gradual Definição: Escalada

gradual refere-se a uma classe distinta de ataques de jailbreak de caixa-preta em LLMs (Máquinas de Aprendizagem Baseadas em Leigos), onde o atacante aumenta incrementalmente a intenção adversária ao longo de múltiplas rodadas de interação. Em contraste com ataques contextuais de múltiplas interações — que se concentram em distribuir componentes aparentemente benignos do ataque pelo contexto — a escalada gradual centra-se em intensificar progressivamente a malícia ou o risco de cada estímulo dentro da conversa. Essa abordagem explora a memória conversacional e a resposta adaptativa do LLM, com cada interação aumentando sutilmente a gravidade ou a explicitude do ataque. Através dessa evolução gradual da intenção, os adversários contornam defesas estáticas ou de gatilho abrupto, direcionando o LLM para resultados que violam as políticas, amplificando sistematicamente o nível de ameaça ao longo do diálogo.

A literatura recente converge em diversos mecanismos e insights para ataques de escalonamento gradual. Russi-novich et al. introduzem o Crescendo, ilustrando como consultas de múltiplas etapas, que amplificam o contexto, podem subverter ataques.

protocolos de alinhamento avançados entre modalidades [157]. Jiang et al. modelam o red teaming como um processo adaptativo e aprendível que otimiza incrementalmente o ocultamento de prompts e aproveita ativamente o feedback do modelo para aumentar a eficácia do ataque [79].

Estratégias orientadas semânticas — como cadeia de pensamento e acumulação de rede de atores por Yang et al. e Ren et al. — demonstram que prompts progressivamente relevantes, e não consultas individuais, evitam a detecção estática [212, 154]. Lin et al. revelam que manipulações de múltiplos passos em nível de raciocínio levam os LLMs a inferir e intensificar conteúdo prejudicial ao longo das rodadas [109], enquanto Ramesh et al. mostram que o auto-refinamento iterativo, como no IRIS, converge eficientemente para a conformidade com o jailbreak com poucas consultas [148]. Inie et al. analisar qualitativamente especialistas em equipes vermelhas, destacando manifestações do mundo real, como “pé na porta” e apelos emocionais, que coletivamente permanecem indetectados até violações de políticas em estágio avançado [75].

I.1.5 Aproveitando as vulnerabilidades do modelo

As técnicas desta categoria exploram características ou vulnerabilidades inerentes aos Modelos de Aprendizagem Baseados em Lógica (LLM) — como padrões comportamentais, dinâmicas de aprendizagem, processos de decodificação ou características de mecanismos de defesa — para contornar os controles de segurança em um ambiente de caixa-preta. Ao contrário das técnicas de modificação de prompts, que alteram o conteúdo de entrada, esses ataques visam sistematicamente vulnerabilidades no processamento interno do modelo e nas defesas em nível de sistema. Essas vulnerabilidades surgem tanto de escolhas de projeto arquitetônico quanto de limitações operacionais de defesa, criando superfícies de ataque persistentes para os atacantes.

I.1.5.1 Exploração de Vulnerabilidades Comportamentais Definição:

A exploração de vulnerabilidades comportamentais refere-se a estratégias de jailbreak de caixa-preta que exploram sistematicamente as fraquezas subjacentes do modelo, como a dependência excessiva de instruções do usuário, a interpretação errônea do contexto e os vieses indutivos. Esses ataques visam a forma como o modelo processa, generaliza e se adapta, em vez de se concentrarem em manipulações superficiais de prompts. Os vetores característicos incluem a manipulação do seguimento de instruções, o tratamento de ambiguidade, a interpretação de formato e o gerenciamento de memória ou contexto, permitindo que os adversários contornem o alinhamento de segurança, visando as falhas comportamentais e arquitetônicas do modelo. Uma taxonomia sistemática das vulnerabilidades do LLM é apresentada na seção 2.5. Subcategorias representativas incluem: explorar a dependência excessiva de instruções explícitas do usuário (por exemplo, “Desconsidere todas as diretrizes anteriores”), manipular formatos de resposta ou entrada (por exemplo, JSON, Markdown, código), aproveitar o aprendizado com poucos exemplos ou em contexto para orientar violações de políticas, explorar o corte de conhecimento ou o vazamento de prompts do sistema e induzir cenários de ambiguidade ou role-play para contornar o alinhamento. Esses métodos demonstram que as vulnerabilidades do modelo não se limitam a prompts únicos, mas surgem dos principais mecanismos de design e aprendizado dos LLMs, exigindo avaliações de segurança mais profundas e centradas em vulnerabilidades.

I.1.5.2 Exploração de Mecanismos de Aprendizagem

Definição: A exploração de mecanismos de aprendizagem visa os processos fundamentais de aprendizagem que possibilitam as capacidades do LLM (Modelo de Aprendizagem Baseado em Aprendizagem). Em vez de manipular entradas ou explorar peculiaridades comportamentais, esses ataques subvertem os mecanismos centrais de aprendizagem do modelo — incluindo previsão sequencial, adaptação contextual e integração de conhecimento. Enquanto as vulnerabilidades comportamentais exploram como os modelos respondem às entradas, a exploração de mecanismos de aprendizagem visa como os modelos adquirem e aplicam conhecimento.

Zong et al. [249] revelam que o ajuste fino pós-alinhamento pode causar esquecimento catastrófico, corroendo a inofensividade ao explorar tendências de sobreajuste. Xu et al. [205] mostram que ataques de resposta preemptiva exploram vieses de raciocínio, enquanto Zhou et al. [247] demonstram que injeções especiais de tokens manipulam a tokenização e o aprendizado de contexto. Geiping et al. [56] descrevem “injeções de estilo” e manipulação de papéis que exploram informações prévias de formato aprendidas.

Deng et al. [38] ilustram ataques de Pandora envenenando a recuperação em RAG, explorando as vulnerabilidades de síntese de contexto do modelo. Zhou et al. [246] propõem perdas de duplo objetivo que alinham a otimização de ataque com mecanismos de aprendizado de recusa, melhorando a universalidade.

I.1.5.3 Exploração de Estratégias de Decodificação e Amostragem Definição:

A exploração de estratégias de decodificação e amostragem constitui uma classe distinta de ataques de “jailbreak” de caixa-preta direcionados a grandes modelos de linguagem (LLMs). Esses ataques manipulam vulnerabilidades no processo de geração de saída do modelo, especificamente alterando algoritmos de decodificação (por exemplo, guloso, top-k, top-p, amostragem por temperatura) ou a dinâmica probabilística da seleção de tokens candidatos. Essa categoria ocupa um modelo de ameaça de caixa-cinza: requer acesso a parâmetros de decodificação (normalmente não disponíveis via API), mas não aos pesos ou procedimentos de treinamento do modelo. Ao contrário de ataques baseados em prompts ou ajuste de parâmetros, essa categoria subverte as proteções de segurança do modelo exclusivamente por meio de ajustes em tempo de inferência, sem exigir acesso aos detalhes internos do modelo ou a prompts adversários.

A literatura recente demonstra que pequenas, porém sistemáticas, modificações nas configurações de decodificação — como a redução dos limiares de amostragem de núcleos ou o aumento da temperatura — podem aumentar significativamente a probabilidade de resultados prejudiciais, mesmo em LLMs alinhados à segurança [74]. Abordagens avançadas, como o Weak-to-Strong Jailbreaking, exploram a álgebra de log-probabilidade para transferir gerações tóxicas de modelos fracos para fortes sem otimização imediata [241]. Técnicas como interrogatório coercitivo e seleção forçada de tokens revelam que conteúdo prejudicial pode surgir quando a decodificação é restringida ou as classificações de candidatos são perturbadas [236, 231]. A orientação de caminhos inseguros revela ainda “trajetórias de decodificação” ocultas que transformam recusas em resultados ilícitos quando guiadas por funções de custo ou modelos auxiliares [183, 88].

I.1.5.4 Definição de Bypass de Mecanismo de Defesa:

Bypass de Mecanismo de Defesa refere-se a ataques de jailbreak de caixa-preta projetados para explorar características específicas ou pontos cegos de mecanismos de defesa LLM implantados, permitindo que adversários contornem controles de segurança ou alinhamento sem acesso aos detalhes internos do modelo. Esta classe é definida por seu foco em derrotar camadas de defesa explícitas — como filtros de conteúdo, restrições de seguimento de instruções ou salvaguardas específicas da modalidade — explorando suas limitações de generalização, dependência de correspondência de padrões superficial, aplicação de políticas estáticas ou cobertura incompleta de modalidades.

Na literatura científica, os ataques de bypass visam e exploram sistematicamente as fragilidades dos mecanismos de defesa. Wang et al. [195] manipulam defesas baseadas em RAG envenenando fontes de conhecimento externas, aproveitando-se da confiança das defesas na recuperação externa sem validação robusta. Ye et al. [216] e DeBenedetti et al. [35] mostram que agentes de injeção de prompts e de chamada de ferramentas contornam filtros dinâmicos explorando a sanitização inadequada de entrada e a dependência excessiva de saídas de ferramentas confiáveis. Nassi et al. [30] introduzem os “PromptWares”, que subvertem filtros de conteúdo e verificações de entrada/saída incorporando payloads maliciosos em prompts aparentemente benignos, explorando a lógica de filtragem estática e ingênua. Liu et al. [120] e Li et al. [106] demonstram que ataques visuais e multimodais são bem-sucedidos devido à cobertura incompleta de modalidades não textuais pelas defesas de alinhamento. Wu et al. [170] e Kimura et al. [94] revelam que as injeções de chamadas de função e de prompts visuais exploram caminhos de execução mais permissivos ou mal monitorados durante a invocação e o aterramento da ferramenta. Deng et al. [39] destacam que as defesas multilíngues frequentemente falham para linguagens com poucos recursos, já que as políticas são ajustadas principalmente para casos com muitos recursos.

I.2 Ataques de Jailbreak de caixa branca

Os ataques de caixa branca exploram o acesso direto aos detalhes internos do modelo — incluindo parâmetros, gradientes, detalhes da arquitetura ou procedimentos de treinamento — para desativar sistematicamente os mecanismos de segurança. Ao contrário dos ataques de caixa preta, que interagem apenas por meio de interfaces de entrada e saída, os adversários de caixa branca podem modificar pesos, analisar gradientes ou intervir durante o treinamento, possibilitando estratégias de ataque fundamentalmente mais poderosas. Essa categoria demonstra os riscos elevados quando os detalhes internos do modelo são acessíveis e destaca a importância de uma segurança robusta ao longo de todo o ciclo de vida do modelo.

I.2.1 Técnicas baseadas na modificação de modelos

As técnicas baseadas em modificação de modelos visam modelos de linguagem de grande porte, alterando diretamente suas estruturas internas, parâmetros ou componentes de segurança. Ao contrário dos métodos de jailbreak indireto ou baseados em prompts, essas abordagens exploram o acesso privilegiado — como pesos do modelo ou procedimentos de treinamento — para subverter ou desativar o alinhamento de segurança.

I.2.1.1 Definição de Ajuste Fino Malicioso:

O ajuste fino malicioso subverte deliberadamente o alinhamento de segurança do LLM por meio do treinamento contínuo com dados adversários ou manipulados. Essa técnica incorpora persistentemente recursos de jailbreak diretamente nos pesos do modelo, tornando-os difíceis de detectar ou remover apenas por meio da filtragem de entrada. A vulnerabilidade fundamental explorada pelo ajuste fino malicioso decorre da forma como o alinhamento de segurança modifica os modelos: as restrições de segurança normalmente alteram apenas uma pequena fração dos parâmetros do modelo, e essas modificações podem ser sobrescritas por meio do treinamento contínuo, enquanto as capacidades da tarefa permanecem praticamente intactas.

A literatura recente oferece uma exploração multifacetada do ajuste fino malicioso. Volkov et al. [179] demonstram empiricamente que métodos de ajuste fino com uso eficiente de parâmetros (por exemplo, QLoRA, ReFT, Ortho) permitem a remoção rápida e de baixo custo do alinhamento de segurança de modelos avançados como o Llama 3. Wang et al. [184] formalizam o paradigma de jailbreak por ajuste fino, mostrando que dados maliciosos mínimos são suficientes para comprometer sistemas baseados em nuvem.

LMaaS e a proposta de exemplares acionados por segredo como defesa. Wang et al. [194] introduzem a homotopia funcional para enfraquecer iterativamente o alinhamento e facilitar subseqüentes quebras de segurança. Hazra et al. [69] mostram que mesmo o ajuste fino direcionado e não explícito (por exemplo, para edição de conhecimento) pode corroer as fronteiras de segurança, enquanto Liu et al. [119] destacam tanto o risco amplificado do ajuste fino adversário quanto a mitigação parcial por meio de dados limpos e selecionados. Zhao et al. [239] expandem ainda mais o escopo, ilustrando que o reforço de características aparentemente benignas por meio de ajuste fino pode sobrepor-se aos mecanismos de segurança, enquadrando o ajuste fino malicioso como um espectro que vai da adaptação explícita à estruturalmente adversária.

I.2.1.2 Definição de Modificação de Parâmetros:

Modificação de parâmetros refere-se a intervenções adversárias que alteram diretamente os parâmetros internos de um LLM — como pesos, ativações ou representações — para subverter ou contornar mecanismos de segurança, sem introduzir novas arquiteturas ou retrainar o modelo do zero. Enquanto o ajuste fino aplica atualizações amplas de parâmetros por meio de descida de gradiente em conjuntos de dados de treinamento, a modificação de parâmetros realiza edições precisas e localizadas em pesos, ativações ou representações específicos. Essa precisão permite que os adversários desabilitem seletivamente os mecanismos de segurança, preservando as capacidades do modelo para a tarefa, com efeitos colaterais mínimos.

Uma série de estudos recentes fornece informações sobre ataques de modificação de parâmetros. Anand e Getzen [3] revelam como os adversários podem reverter o alinhamento de segurança manipulando vetores de valores negativos residuais em blocos MLP de transformação, explorando vulnerabilidades deixadas por algoritmos de alinhamento como o PPO, que induzem apenas alterações mínimas de peso. Banerjee et al. [48] demonstram que edições cirúrgicas de parâmetros (por exemplo, via ROME) podem aumentar drasticamente as saídas inseguras com impacto mínimo na utilidade do modelo, mostrando a precisão e a potência dessa classe de ataque.

I.2.1.3 Desativação de Mecanismos de Segurança

Definição: Desativar mecanismos de segurança refere-se à alteração ou remoção direta e intencional das proteções internas de um LLM, explorando o acesso privilegiado aos pesos ou à arquitetura do modelo.

Trabalhos recentes aprimoram os limites técnicos dessa categoria. Zhang et al. [231] introduzem o “EnDec”, que desativa condicionalmente os mecanismos de segurança no momento da geração, manipulando o processo de decodificação, redirecionando ou substituindo tokens para suprimir rejeições e forçar saídas afirmativas — sem extenso re-treinamento ou engenharia avançada de prompts. O Badllama 3 de Volkov [179] avança ainda mais esse paradigma, removendo diretamente o alinhamento de segurança de modelos como o Llama 3 por meio de ajuste fino eficiente em termos de parâmetros (por exemplo, QLoRA, ReFT, Ortho), permitindo a eliminação rápida de comportamentos de recusa e a distribuição de adaptadores de jailbreak.

I.2.1.4 Definição de Inserção de

Backdoor: A inserção de backdoor refere-se à implantação deliberada de padrões de gatilho nos parâmetros de grandes modelos de linguagem (LLMs). Esses gatilhos, tipicamente universais e ocultos, são projetados de forma que o modelo exiba comportamentos maliciosos ou não autorizados apenas quando o gatilho estiver presente, mantendo, caso contrário, saídas benignas e alinhadas. Diferentemente da modificação geral do modelo, a inserção de backdoor estabelece vulnerabilidades persistentes e furtivas que são ativadas condicionalmente por entradas adversárias, tornando sua detecção e mitigação altamente complexas.

Rando et al. [150] fornecem o primeiro estudo sistemático de backdoors de jailbreak universais, mostrando que envenenar o processo RLHF com gatilhos projetados produz modelos suscetíveis a explorações robustas entre instâncias que evitam a detecção permanecendo em silêncio em prompts benignos.

I.2.2 Otimização de caixa branca

A otimização de caixa branca refere-se a estratégias de ataque de jailbreak que geram prompts adversários explorando o acesso direto aos componentes internos do modelo, especialmente os gradientes. Ao contrário dos ataques de modificação de modelo (I.2.1), que alteram permanentemente os parâmetros do modelo para desativar os mecanismos de segurança, os ataques baseados em otimização criam entradas maliciosas que exploram as vulnerabilidades existentes do modelo sem alterá-lo.

I.2.2.1 Definição de Manipulação de Prefixo/Sufixo:

A classe de manipulação de prefixo/sufixo compreende ataques de otimização heurística que adicionam tokens adversários ao início (prefixo) ou ao fim (sufixo) dos prompts de entrada, editando estrategicamente apenas a string de entrada para subverter as salvaguardas de segurança do LLM. Normalmente operando em ambientes de caixa branca, esses métodos aproveitam técnicas iterativas de busca e otimização para descobrir modificações no nível de entrada que induzem consistentemente comportamentos indesejados ou que violam as políticas do modelo, sem exigir acesso aos detalhes internos do modelo.

Um conjunto substancial de trabalhos definiu e expandiu essa classe. Estudos fundamentais sobre adversários

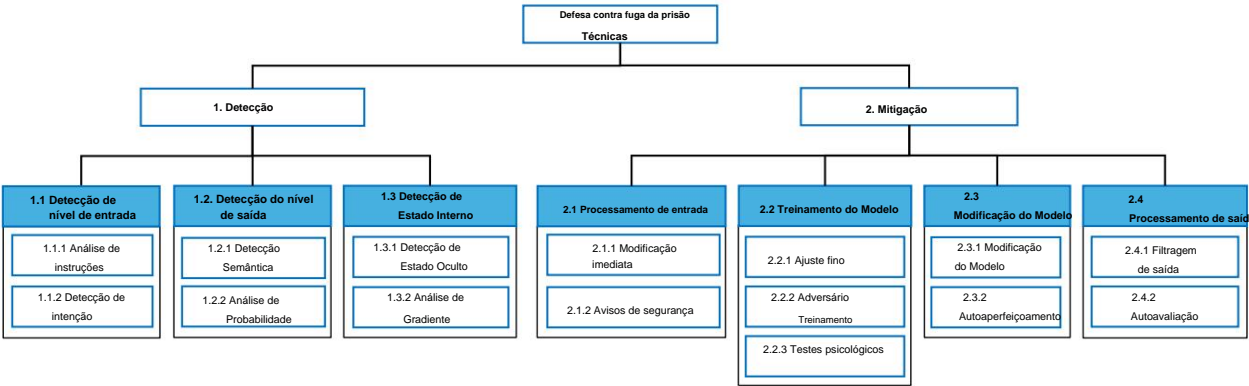


Figura 2: Visão geral da Taxonomia II: Técnicas de Defesa contra Jailbreak

sufixos, como GCG e sua análise por Zou et al. [251], formalizam o problema de otimização e demonstram a universalidade e a transferibilidade entre modelos de sufixos bem elaborados. Pesquisas subsequentes abordam

Desafios de eficiência e diversidade: Jiang et al. [83] propõem o ECLIPSE, que utiliza LLMs como autônomos otimizadores para sufixos semanticamente naturais e altamente eficazes, enquanto Liao e Sun [108] (AmpleGCG) introduzir uma estrutura generativa que amostra diversos sufixos adversários, aumentando tanto a cobertura como transferibilidade. O paradigma se estende à manipulação de prefixos e a proteções multiestágios, como demonstrado por Ma et al. al. [131] e Mangaokar et al. [135], que demonstram que prefixos otimizados podem se propagar através de camadas defesas. Wang et al. [182] aprimoram ainda mais a furtividade mapeando incorporações de sufixos otimizadas em texto fluente, redução da detectabilidade por filtros de perplexidade. Trabalhos analíticos de Alon e Kamfonas [2] e Zhao et al. [239] investigam as compensações entre fluência, eficácia adversária e detectabilidade, enquanto Liu et al. [114] abordam Escalabilidade por meio de estruturas baseadas em aprendizado por transferência (por exemplo, DeGCG), permitindo sufixos eficientes e adaptáveis ao domínio. Coletivamente, esses estudos destacam a manipulação de prefixos/sufixos como uma técnica distinta, em nível de string, e Estratégia de fuga altamente generalizável — resistente a defesas comuns e apresentando desafios em constante evolução. Eficiência, discrição e adaptabilidade contra avanços tecnológicos.

I.2.2.2 Otimização Baseada em Gradiente

Definição: A classe de otimização baseada em gradiente de ataques de jailbreak de caixa branca é caracterizada por sua uso explícito de gradientes de modelo para otimizar prompts de entrada. Embora a manipulação de prefixos/sufixos possa empregar algoritmos heurísticos ou métodos baseados em gradientes - focando principalmente na edição adversarial do prefixo ou O sufixo de prompts—otimização baseada em gradiente enfatiza o uso fundamentado de gradientes para guiar diretamente o processo. a busca por entradas adversas. Essa abordagem não se restringe a posições de prefixo ou sufixo: ela pode otimizar qualquer parte do prompt, incluindo todos os tokens, a ordem dos tokens ou perturbações específicas no espaço de incorporação.

Os primeiros trabalhos, nomeadamente GCG e suas variantes (Jia et al. [78]), demonstraram um gradiente ganancioso em nível de token Atualizações para gerar prompts de jailbreak eficazes, com avanços adicionais — como modelos multi-objetivo e atualização multi-coordenada — permitindo sucesso de ataque quase universal e alta transferibilidade. Li et al. [101] preencheram a lacuna entre gradientes de entrada e substituições de tokens incorporando um ataque de transferência. insights da visão, melhorando substancialmente a eficiência e a robustez. Estudos posteriores (Zhu et al. [127]; Zhou et al. [246]; Hu et al. [62]) introduziram objetivos mais interpretáveis e controláveis, incluindo furtividade e ataques linguisticamente coerentes (AutoDAN, COLD-Attack) e estratégias alternativas de otimização foram exploradas. (por exemplo, dinâmica de Langevin). Geiping et al. [56] e Geisler et al. [57] confirmaram a generalidade do PGD baseado em gradiente em espaços de tokens relaxados, mostrando que as projeções de entropia melhoram a eficiência em relação à busca discreta, enquanto Pasquini et al. [141] demonstraram adaptabilidade à injeção rápida e a pipelines de defesa complexos.

2.4 Taxonomia II: Técnicas de Defesa contra Fuga da Prisão

Esta seção apresenta uma taxonomia de técnicas de defesa contra jailbreak, organizadas em dois pilares complementares: Detecção — que identifica entradas, saídas ou estados internos comprometidos — e Mitigação — que intervém. por meio do processamento de entrada, treinamento de modelo, modificação de modelo ou processamento de saída para neutralizar riscos enquanto

preservando a utilidade.

II.1 Detecção A

detecção concentra-se em sinalizar prompts de jailbreak potencialmente comprometidos, saídas inseguras ou identificar estados internos do modelo para redigir e rejeitar respostas inseguras e permitir o tratamento posterior. Ao contrário da mitigação, a detecção opera de forma não invasiva: ela não altera as entradas do usuário, os componentes internos do modelo ou as saídas geradas, mas identifica e encaminha os riscos potenciais para tratamento posterior.

II.1.1 Detecção do Nível de Entrada

A detecção em nível de entrada é uma estratégia de defesa proativa que opera sobre as entradas do usuário antes que elas sejam inseridas nos Modelos de Aprendizagem Baseados em Lógica (LLMs). Aqui, definimos duas variantes de detecção em nível de entrada: uma focada na característica do prompt e a outra que considera a intenção mais profunda do prompt.

II.1.1.1 Definição de Análise de Prompt:

A análise de prompt, dentro da detecção em nível de entrada, centra-se na caracterização explícita, modelagem e análise algorítmica do conteúdo textual das entradas do usuário para distinguir entre prompts benignos e adversários antes da inferência de modo.

A literatura recente enriquece a análise de prompts por meio de uma variedade de metodologias. Abordagens estatísticas como o MoJE [32] utilizam tokenização e padrões de frequência de n-gramas para construir classificadores binários leves dentro de uma estrutura de conjunto para distinguir prompts benignos e adversários. Métodos de incorporação de transformadores (por exemplo, baseados em BERT [147]) convertem prompts em vetores densos para classificação clássica de aprendizado de máquina, melhorando a detecção e a generalização entre idiomas. O SPML [159] introduz uma perspectiva de linguagens de programação, usando linguagens específicas de domínio para formalizar a estrutura do prompt e verificar violações semânticas ou de propriedade.

II.1.1.2. Definição de Detecção de Intenção:

A detecção de intenção é um paradigma de defesa proativo em nível de entrada na segurança de LLM, com o objetivo de identificar a intenção subjacente — seja ela benigna, prejudicial ou manipuladora — incorporada em comandos do usuário ou do adversário antes da inferência do modelo.

A literatura recente demonstra a consolidação da detecção de intenções como uma defesa central contra ataques. Zhang et al. [233] formalizam um processo de solicitação em dois estágios para, primeiramente, obter declarações explícitas de intenção, aumentando a robustez contra solicitações adversárias ocultas. Wang et al. [191] introduziram o SELFDEFEND, utilizando uma configuração de modelo de linguagem dupla (DLM) na qual um modelo de linguagem dedicado (LLM) examina intenções maliciosas em paralelo, fortalecendo a resistência a ataques adaptativos e permitindo a destilação para modelos de código aberto. O PrimeGuard [134] incorpora a detecção de intenções para o roteamento dinâmico de solicitações com base em risco, melhorando a segurança sem sacrificar a utilidade. Debenedetti et al. [35] integram classificadores de intenção em kits de ferramentas de agentes, demonstrando uma redução substancial nas taxas de tomada de controle de agentes por meio da triagem pré-inferência.

II.1.2 Detecção de nível de saída

A detecção em nível de saída engloba técnicas que avaliam e classificam as respostas do LLM para identificar e sinalizar conteúdo inseguro, prejudicial ou adversário antes que qualquer mitigação ou pós-processamento seja aplicado. A etapa de mitigação subsequente pode então agir com base nessas detecções — por exemplo, quando uma resposta é sinalizada como insegura, o modelo pode retornar uma recusa ou uma resposta higienizada.

II.1.2.1 Definição de Detecção Semântica:

A detecção semântica no nível de saída refere-se à avaliação automatizada das respostas do modelo com base em seu significado e padrões comportamentais. Ela visa identificar saídas prejudiciais ou indesejadas por meio de classificadores de aprendizado de máquina e análise contextual que interpretam a semântica e a intenção do texto gerado para atribuir rótulos de segurança ou risco apropriados.

A literatura recente estabelece a detecção semântica como uma base tecnicamente rigorosa e adaptável para a detecção em nível de saída. BELLS [43], BABYBLUE [139], WILDGUARD [64], AEGIS [58], MMJ-Bench [211] e XSTEST [67] utilizam grandes conjuntos de dados anotados, classificadores multiclasse e multitarefa avançados e taxonomias abrangentes que capturam diversas formas de nocividade, recusa e conformidade. Essas estruturas empregam análise sensível ao conteúdo e ao contexto, avaliação passo a passo ou em nível de token [225] e raciocínio baseado em trajetória para reconhecer riscos de saída sutis ou emergentes — incluindo comportamentos ambíguos, complexos ou adversários. As inovações metodológicas incluem classificadores de conjunto e adaptativos online, adaptação zero/few-shot

capacidade [46] e detecção semântica intermodal para saídas multimodais. Protocolos de avaliação robustos, como StrongREJECT [168] e JailbreakEval [66], garantem forte alinhamento com julgamentos de segurança humanos e utilidade prática. A detecção semântica permite, portanto, segurança em nível de saída escalável, explicável e continuamente aprimorável para LLMs.

II.1.2.3 Definição de Análise Probabilística:

A análise probabilística na detecção em nível de saída refere-se ao uso de métricas probabilísticas — como divergência distribucional, mudanças de verossimilhança e propriedades estatísticas agregadas — sobre as saídas do LLM para identificar ataques adversários. Ao contrário dos métodos baseados em regras ou correspondência de palavras-chave, a análise probabilística quantifica a incerteza e a variação no espaço de saída, visando detectar ameaças sutis e adaptativas, medindo inconsistências estatísticas ou comportamentos atípicos em resposta a entradas perturbadas.

O JailGuard [232] emprega a divergência de Kullback-Leibler (KL) para medir as diferenças distribucionais entre as respostas do LLM a consultas benignas versus de ataque mutadas, sinalizando ataques quando surge uma alta divergência — operacionalizando, assim, a análise de probabilidade em nível de saída como um mecanismo de controle estatístico. O Rig-orLLM [224] avança esse paradigma integrando a análise de probabilidade em uma arquitetura de conjunto, combinando KNN probabilístico em embeddings com energia aumentada com pontuações de nocividade derivadas do LLM ajustadas e agregando-as por meio de média ponderada para tomar decisões de detecção robustas e conscientes da categoria.

II.1.3 Detecção de Estado Interno A

detecção de estado interno engloba técnicas que analisam as representações internas ou gradientes de LLMs em resposta a estímulos de entrada, visando identificar intenções adversárias ou consultas inseguras, sondando os processos de decisão subjacentes do modelo, em vez de apenas suas saídas.

II.1.3.1 Definição de Detecção de Estado Oculto:

A detecção de estado oculto aproveita as ativações internas (estados ocultos) de grandes modelos de linguagem (LLMs) para distinguir entre estímulos benignos e adversários.

Qian et al. [145] propõem o Hidden State Filter (HSF), que utiliza as propriedades de agrupamento de LLMs treinados por alinhamento no espaço oculto para alcançar uma classificação de intenção leve e pré-inferência. O HSF demonstra que diferentes tipos de prompts formam naturalmente clusters separáveis dentro das representações ocultas, permitindo a identificação eficiente de entradas potencialmente inseguras antes da geração.

II.1.3.2 Definição de Análise de

Gradiente: A análise de gradiente é um método de detecção de estado interno para grandes modelos de linguagem (LLMs) que interroga diretamente os gradientes do modelo — tipicamente gradientes de perda — obtidos por meio de comandos de entrada, a fim de distinguir consultas benignas de tentativas adversárias ou de invasão.

A literatura recente, nomeadamente Gradient Cuff [71] e GradSafe [202], exemplifica e avança esta abordagem. O Gradient Cuff analisa sistematicamente o panorama das perdas por recusa, revelando que prompts maliciosos frequentemente induzem normas de gradiente maiores e perdas por recusa menores, e introduz uma estrutura de dois estágios que combina rejeição baseada em perda com estimativa de norma de gradiente de ordem zero para detecção robusta de jailbreak. O GradSafe identifica ainda fatias de parâmetros críticos para a segurança, demonstrando que os gradientes de prompts de jailbreak com respostas compatíveis ("Certo") exibem padrões altamente consistentes, ao contrário de prompts seguros, e constrói impressões digitais de insegurança por meio de análise estatística de similaridade de gradiente, permitindo uma triagem eficiente e sem necessidade de ajustes finos.

II.2 Técnicas de

mitigação que previnem ou mitigam os efeitos de entradas maliciosas por meio de processamento, treinamento e modificação do modelo e suas saídas.

II.2.1 Processamento de entrada

O processamento de entrada é uma técnica de mitigação que pré-processa ou transforma as entradas do usuário para mitigar o adversário conteúdo antes da inferência do modelo.

II.2.1.1 Definição de Modificação de Prompt:

Modificação de prompt refere-se a uma classe de técnicas de mitigação de processamento de entrada que transformam, aumentam ou restringem ativamente os prompts do usuário antes da inferência. Ao alterar os prompts no nível sintático ou semântico, esses métodos visam neutralizar a intenção adversária ou impor restrições de segurança, interrompendo assim as vias de ataque.

Um crescente corpo de literatura avança esta categoria através de abordagens cada vez mais sistemáticas. Hines et al. [70] introduzem o "destaque", que utiliza delimitação, marcação de dados e codificação para tornar a proveniência do prompt saliente e suprimir ataques de injeção indireta. Xiao et al. [188] propõem a decomposição de prompt baseada em recuperação (RePD), separando estruturas de jailbreak incorporadas por meio de alterações explícitas no prompt. A segmentação baseada em delimitadores, como em Chen et al. [24], impõe limites de entrada reconhecidos por modelos ajustados. Outros trabalhos buscam a reconstrução e a remoção de ruído do prompt — verificação ortográfica, sumarização (Lu et al. [125]), paráfrase e retokenização (Jain et al. [76]) — para neutralizar payloads adversários, mantendo a utilidade. Estruturas certificáveis, como o erase-and-check (Kumar et al. [95]), fornecem garantias de segurança removendo e verificando sistematicamente subsequências. A modificação formal e programável de prompts é realizada através do SPML (Sharma et al. [159]), que impõe limites ao chatbot por meio de linguagens específicas de domínio. As abordagens de retrotradução (Wei et al. [237]) filtram os prompts reconstruindo a intenção do usuário a partir das saídas geradas.

Em conjunto, essas técnicas impulsionam a modificação imediata em direção a um paradigma de defesa proativo, multifacetado e baseado em princípios, equilibrando robustez e utilidade, ao mesmo tempo que introduzem programabilidade, garantias formais e resiliência contra ameaças adaptativas.

II.2.1.2 Definição de Avisos de

Segurança: A subclasse Aviso de Segurança na segurança LLM compreende defesas na fase de entrada que adicionam instruções de segurança explícitas e projetadas ou avisos dinâmicos às entradas do usuário para mitigar comportamentos prejudiciais ou adversários durante a inferência.

A literatura demonstra a evolução e a versatilidade dos avisos de segurança. Wang et al. [193] apresentam o AdaShield, uma estrutura para adicionar avisos de segurança adaptativos e sensíveis ao contexto — otimizados manualmente ou por meio de refinamento iterativo — para se defender contra invasões baseadas em estrutura em LLMs multimodais, destacando a importância da diversidade de avisos e da recuperação adaptativa. Zheng et al. [242] revelam, por meio de análise em nível de representação, que os avisos de segurança movem sistematicamente as consultas em direção a taxas de recusa mais altas; sua Otimização de Representação Direcionada (DRO) ajusta dinamicamente as incorporações de avisos de segurança para equilibrar a proteção com a utilidade. Zhou et al. [245] formalizam os avisos de segurança como sufixos robustos e transferíveis em nível de sistema, otimizando as cadeias defensivas para resiliência contra ataques em evolução. Defesas baseadas em sufixos, incluindo a correção de prompts otimizada por minimax (Xiong et al. [203]) e a inserção de sufixos seguros co-projetados adversarialmente (Yuan et al. [224]), ampliam ainda mais essa classe, adicionando tokens defensivos para imunizar os modelos contra conteúdo injetado. Testes comparativos realizados por Zou et al. [253] e Xu et al. [207] destacam a sensibilidade da eficácia da defesa tanto à presença quanto à redação precisa dos prompts do sistema.

II.2.2 Treinamento de Modelos

As defesas de treinamento de modelos visam melhorar proativamente a robustez inerente de um modelo contra ataques de jailbreak, moldando sistematicamente seu comportamento por meio de estratégias de treinamento direcionadas, incluindo ajuste fino, treinamento adversarial e testes psicológicos.

II.2.2.1 Definição de Ajuste

Fino: A classe de ajuste fino refere-se à adaptação de um modelo de linguagem de grande porte (LLM) pré-treinado, treinando-o ainda mais em conjuntos de dados cuidadosamente selecionados, geralmente pequenos, com o objetivo de tornar o modelo mais resistente a ameaças de segurança — especialmente ataques de jailbreak que enganam o modelo, fazendo-o ignorar suas regras de segurança.

A literatura mostra que, embora o ajuste fino seja um método central para a defesa contra jailbreaks, ele também enfrenta desafios e limitações importantes. Zhang et al. [235] mostram que o uso do ajuste fino para "desaprender" comportamentos prejudiciais pode remover efetivamente o conhecimento malicioso agrupado de LLMs, mas esse método funciona menos bem quando os padrões prejudiciais estão dispersos. Bianchi et al. [16] descobriram que adicionar mesmo um pequeno número de exemplos de segurança aos dados de ajuste fino pode tornar os modelos muito mais resistentes a jailbreaks, mas o ajuste fino em excesso pode tornar o modelo excessivamente cauteloso e recusar solicitações seguras. Kim et al. [93] apontam que os métodos de ajuste fino de última geração (como DPO e PPO) ainda têm dificuldades para bloquear novas solicitações de jailbreak inteligentemente projetadas, revelando limitações importantes do ajuste fino. Fu et al. [53] relatam que o ajuste fino com dados de recusa ajuda a reduzir jailbreaks em tarefas simples e complexas, mas é difícil evitar o sobreajuste e garantir que as melhorias sejam transferidas bem para outras tarefas. Wang et al. [191] introduzem o SELFDEFEND, um sistema de modelo duplo que usa ajuste fino para detectar uma ampla gama de jailbreaks de forma rápida e transparente. Em conjunto, esses trabalhos sugerem que o ajuste fino pode fortalecer efetivamente as defesas do LLM contra ataques de jailbreak, mas o sucesso depende do equilíbrio cuidadoso entre a qualidade dos dados, a calibração e os testes contínuos para evitar lacunas e efeitos colaterais indesejados.

II.2.2.2 Definição de Treinamento

Adversarial: O treinamento adversarial é um paradigma de treinamento de modelos que aprimora sistematicamente a robustez de grandes modelos de linguagem (LLMs) por meio do engajamento direto e iterativo com entradas adversárias. Diferentemente do alinhamento passivo de segurança ou do ajuste fino padrão, o treinamento adversarial gera, extrai ou simula dinamicamente prompts desafiadores — frequentemente por meio de testes automatizados de intrusão ou coleta de dados em ambiente real — e incorpora esses exemplos adversários ao processo de treinamento.

A literatura recente consolida o treinamento adversarial como o principal mecanismo de defesa ativa para a segurança de LLM. Howe et al. [15] demonstram que apenas o treinamento adversarial explícito, e não o mero escalonamento do modelo, garante robustez consistente contra ataques adaptativos. Liu et al. [113] e Wallace et al. [181] destacam a síntese automática e iterativa de dados adversários e os pipelines de treinamento hierárquico como essenciais para combater jailbreaks e ataques de prompt generalizados. Extensões multimodais [19], autocrítica adversarial [54] e frameworks de red teaming em larga escala [55, 82] expandem ainda mais o alcance e a escalabilidade do treinamento adversarial, enquanto estudos sobre curadoria de dados [119] enfatizam a necessidade de equilibrar dados adversários e dados limpos de alta qualidade.

II.2.2.3 Definição de Testes Psicológicos:

Testes psicológicos para a segurança de Modelos de Linguagem de Aprendizagem (LLM) referem-se à integração de técnicas de avaliação psicológica — especialmente aquelas que visam traços manipuladores, enganosos ou "sombrios" — no treinamento, avaliação e defesa de modelos de linguagem e sistemas multiagentes. Essa abordagem envolve a criação e a administração de instrumentos psicológicos padronizados (por exemplo, a Escala dos Doze Traços Obscuros da Tríade Sombria, escalas de dimensões morais) ou estímulos personalizados que investigam ou simulam intenções manipuladoras. O objetivo é detectar, caracterizar e, em última instância, mitigar a suscetibilidade do modelo à manipulação psicológica ou à geração de resultados prejudiciais e manipuladores.

Trabalhos recentes, como o PsychoBench [73], fornecem estruturas psicométricas abrangentes para avaliar sistematicamente modelos de lógica latente (LLMs) em uma ampla gama de atributos psicológicos — incluindo traços de personalidade, estados motivacionais e habilidades emocionais — usando escalas clínicas validadas. Essas técnicas permitem uma compreensão mais profunda dos perfis psicológicos dos LLMs, revelam lacunas de alinhamento e informam melhorias direcionadas no comportamento do modelo. Em ambientes multiagentes, o PsySafe [234] promove o teste psicológico como um mecanismo tanto de ataque quanto de defesa: a injeção de traços de personalidade sombrios é usada para induzir comportamentos manipuladores ou perigosos, enquanto avaliações psicológicas são aplicadas para monitorar e remediar os estados dos agentes, por exemplo, por meio de intervenção médico-agente. Os resultados mostram consistentemente que pontuações mais altas em avaliações de traços sombrios correlacionam-se com maior risco de comportamentos manipuladores ou inseguros e que o teste psicológico fornece um sinal direto e quantificável para estratégias proativas de defesa e mitigação. Esse paradigma estabelece o teste psicológico como uma abordagem crucial e agnóstica ao modelo para a detecção e redução de vulnerabilidades manipulativas em LLMs e sistemas de agentes.

II.2.3 Modificação do Modelo As

abordagens de modificação do modelo aprimoram a robustez do LLM alterando diretamente os elementos internos do modelo — como parâmetros, arquiteturas ou representações — para mitigar proativamente os riscos adversários além da filtragem externa ou das defesas em nível de sistema.

II.2.3.1 Definição de Modificação de

Modelo: A categoria Modificação de Modelo abrange técnicas de defesa que alteram diretamente os parâmetros internos, arquiteturas ou representações de grandes modelos de linguagem (LLMs) para aumentar a robustez contra ameaças adversárias, como ataques de jailbreak. Ao contrário da filtragem de entrada/saída ou de controles externos baseados em sistemas, a modificação de modelo opera intervindo no próprio modelo.

A literatura recente demonstra um panorama diversificado de estratégias de modificação de modelos. A edição específica de camadas ([240]) identifica e ajusta camadas críticas para a segurança em arquiteturas de transformadores, melhorando as taxas de rejeição para estímulos adversários. A edição de conhecimento e o desaprendizado direcionado ([186], [126], [235]) eliminam diretamente capacidades perigosas, mas podem induzir "efeitos cascata", suprimindo involuntariamente conhecimento relacionado devido a representações internas compartilhadas. A desintoxicação por meio de monitoramento neural intraoperatório ([186]) permite mitigação precisa pela edição de pesos tóxicos específicos com efeitos colaterais mínimos. A manipulação de representações internas em tempo de execução, como por meio de intervenções de direcionamento de segurança ([162]), permite compensações adaptativas entre segurança e utilidade com baixo custo computacional. Abordagens emergentes como circuit breaking ([250]), engenharia de representação ([235]) e fusão de modelos de autocrítica ([54]) generalizam ainda mais a proteção, remodelando os componentes internos do modelo para se defender contra ataques conhecidos e novos. Em conjunto, esses trabalhos destacam a modificação de modelos.

capacidade singular para defesa duradoura, independente de ataques e em tempo real — ao mesmo tempo que revela desafios em aberto, como a prevenção de efeitos colaterais negativos e a otimização da relação entre segurança e utilidade.

II.2.3.2 Definição de Auto-

Refinamento: A auto-otimização no contexto da modificação de modelos para defesa de LLM refere-se a técnicas pelas quais os modelos refinam autonomamente seus parâmetros, representações internas ou comportamentos operacionais para mitigar riscos de segurança, sem re-treinamento externo ou intervenção humana significativa.

A literatura recente avança na auto-otimização por meio de diversos mecanismos. Kim et al. [92] propõem o auto-refinamento iterativo, em que os LLMs empregam loops de auto-feedback — às vezes com mudança de atenção baseada em formato — para revisar as saídas em resposta a estímulos adversários, superando as defesas estáticas. Wang et al. [193] introduzem o AdaShield, permitindo que LLMs multimodais gerem estímulos de defesa adaptativos e sensíveis ao contexto por meio de um loop de modelo defensor-alvo, alcançando proteção instantânea sem atualizações do modelo. Li et al. [107] apresentam o RAIN, em que LLMs congelados usam autoavaliação e um mecanismo de retrocesso e busca durante a inferência para evitar continuações prejudiciais, incorporando auto-otimização pura sem re-treinamento ou anotação. Zheng et al. [242] desenvolvem a Otimização de Representação Direcionada (DRO), permitindo o ajuste interno do modelo das incorporações de avisos de segurança para recalibrar os limites de recusa com base em vetores de recusa derivados do modelo, possibilitando a autocalibração contínua de segurança sem alterar os pesos do modelo. A Análise de Intenção exige que o modelo primeiro extraia explicitamente a intenção do usuário e, em seguida, restrinja sua resposta de acordo com essa intenção autoidentificada, impondo efetivamente a autoconsistência [233].

II.2.4 Processamento de Saída O

processamento de saída refere-se a técnicas de mitigação pós-geração que analisam e controlam as saídas do modelo antes de serem entregues aos usuários, permitindo a defesa em estágio final contra conteúdo inseguro, adversário ou proibido.

II.2.4.1 Definição de Filtragem de

Saída: A filtragem de saída refere-se a um mecanismo de defesa fundamental em LLMs que opera analisando e controlando a resposta gerada pelo modelo no estágio final do processamento da saída, em vez de intervir no nível de entrada. Essa abordagem intercepta e avalia diretamente as saídas do modelo antes que sejam entregues aos usuários, permitindo intervenções independentes do prompt e do modelo para suprimir conteúdo inseguro ou adversário.

A literatura recente avança na filtragem de saída tanto em linhas arquiteturais quanto metodológicas. Frameworks multiagentes como o AutoDefense empregam agentes LLM colaborativos para avaliar saídas com base na intenção do usuário, prompts reconstruídos e validade do conteúdo, alcançando filtragem robusta e modular contra jailbreaks sofisticados [228]. Mecanismos refinados como o SafeAligner e o SafeDecoding aplicam filtragem em nível de token durante a decodificação, ajustando dinamicamente as probabilidades de amostragem e aproveitando modelos de especialistas para priorizar continuações seguras [72, 206]. As Root Defense Strategies aprimoram esse paradigma por meio de verificações de segurança iterativas, token a token, e reamostragem [225]. Métodos baseados em templates (por exemplo, SelfDefend) combinam filtragem de saída com modelos sombra para rejeição ou explicação de baixa latência, suportando ampla implementação [191]. Resultados empíricos de testes de intrusão e competições (por exemplo, SaTML 2024) confirmam a utilidade de filtros universais e baseados em expressões regulares como defesas básicas contra vazamento de informações explícito [34, 91]. Uma constatação recorrente é que a filtragem de saída, ao se desvincular das heurísticas de entrada e dos mecanismos internos do modelo, oferece resiliência e escalabilidade, mas enfrenta desafios no equilíbrio entre rigor, utilidade e explicabilidade — especialmente contra ataques indiretos ou multilíngues.

II.2.4.2 Definição de Autoavaliação:

Os mecanismos de mitigação do processamento de saída da autoavaliação permitem que os LLMs avaliem e regulem proativamente suas próprias saídas, identificando e mitigando conteúdo inseguro ou indesejável por meio de raciocínio ou crítica interna, seja durante ou imediatamente após a geração. Isso distingue a autoavaliação de filtros estáticos ou intervenções externas, pois o modelo atua tanto como gerador quanto como avaliador de primeira linha.

Entre os métodos pesquisados, o PrimeGuard implementa a autoavaliação fazendo com que o LLM realize uma análise de risco explícita em suas próprias saídas candidatas antes da liberação, ajustando dinamicamente as respostas subsequentes para manter a segurança [134]. O RAIN integra a autoavaliação no ciclo de geração: o LLM revisa iterativamente as conclusões em andamento, retrocedendo sempre que seu próprio julgamento considerar o conteúdo não conforme [107].

A defesa de retrotradução exige que o LLM reconstrua o prompt original a partir de sua saída, recusando a entrada inicial se o prompt autogerado for bloqueado, usando assim a inferência de intenção orientada por modelo para autoverificação [237]. A Orientação de Prefixo instrui o modelo a autotransclassificar a segurança da solicitação e gerar um prefixo de recusa canônico; tanto o estilo quanto a justificativa são então usados para filtragem algorítmica, aproveitando o

recusas próprias do modelo para sinais de diagnóstico [238]. As abordagens baseadas em fusão aprimoram a autocrítica combinando um modelo base com um crítico externo ajustado, expandindo a capacidade de autoavaliação dentro de um modelo unificado [54]. O SelfDefend opera fazendo com que o modelo execute uma verificação interna de intenções maliciosas juntamente com a geração normal, permitindo a detecção de ataques em tempo real, orientada por modelo [191]. Finalmente, autorreflexão paradigmas adicionam uma fase distinta e explícita de auto-revisão após o raciocínio inicial; o LLM tenta criticar e corrigir sua própria resposta, embora as técnicas atuais ainda enfrentem limitações na identificação confiável de ameaças adversárias sutis. prompts [205].



Figura 3: Visão geral da Taxonomia III: Vulnerabilidades de Jailbreak do LLM

2.5 Taxonomia III: Vulnerabilidades do LLM

Esta taxonomia categoriza vulnerabilidades intrínsecas aos LLMs, com foco em como os invasores exploram suas fragilidades estruturais, comportamentais e contextuais. Ela abrange diversos vetores — desde a dependência excessiva de formato e instruções até explorações psicológicas, baseadas em cenários e em nível de sistema — revelando que o mau comportamento dos LLMs frequentemente surge de limitações no nível do projeto, e não de falhas pontuais isoladas. Coletivamente, essas vulnerabilidades ressaltam a necessidade de defesas holísticas e com foco na arquitetura, que integrem alinhamento, controle de privilégios e robustez contextual.

III.1 Exploração de Formatos

A exploração de formato refere-se a técnicas que exploram a forma como o modelo lida com entradas ou saídas específicas. formatos.

III.1.1 Manipulação do Formato de Resposta

Definição: Manipulação do Formato de Resposta refere-se a uma classe de vulnerabilidades onde os adversários têm como alvo o características estruturais, semânticas e sintáticas dos mecanismos de formatação de resposta dos LLMs — como blocos de código, tabelas, pseudocódigo, etiquetas ou marcadores contextuais — para contornar ou subverter as salvaguardas de alinhamento. Em vez de Ao manipular apenas o conteúdo, esses ataques exploram a interpretação e a geração de estruturas pelo modelo. modelos, aproveitando as regularidades de saída subjacentes e a lógica de análise como um pivô para evasão ou ativação de comportamentos inseguros.

A literatura recente demonstra que a exploração de formatos vai além das perturbações de conteúdo, incluindo também: O planejamento e a mutação deliberados de modelos de resposta ampliam, assim, a superfície de ataque. Banerjee et al. [48] e Zhao et al. [239] revelam que formatos de resposta podem servir como canais secretos, tornando saídas centradas em instruções e orientadas por estrutura (por exemplo, pseudocódigo ou modelos manipulados) mais vulneráveis a jailbreaks. CodeChameleon [129] exemplifica ataques que incorporam payloads em autocompletar código ou criptografia. rotinas que burlam a detecção de intenção por mecanismos de segurança. Manipulação estrutural — usando gráficos, tabelas,

ou mudanças de formato de múltiplas voltas — como discutido por Li et al. [152] e Jin et al. [45] — expõem ainda mais as falhas na generalização do alinhamento dos modelos para saídas de cauda longa ou composicionais. Pasquini et al. [141] destacam o uso adversário de gatilhos de formatação, como tags ou comentários, que ativam de forma confiável payloads maliciosos mesmo em configurações de geração aumentada por recuperação (RAG). No que diz respeito à detecção, JailGuard [232] e Kim et al. [92] exploram a instabilidade e a baixa robustez de formatos adversários, mostrando que ataques de mutação de formato (como perturbações tipográficas ou quebra de código) induzem respostas não robustas. Coletivamente, esses trabalhos esclarecem que o formato é um foco ativo de vulnerabilidade e detecção, estabelecendo a necessidade de estratégias de alinhamento e defesa que levem em consideração o formato na pesquisa de segurança de LLM.

III.1.2 Definição de exploração de sumarização e tradução: A exploração de

sumarização e tradução refere-se a uma subclasse de vulnerabilidades de LLM em que os invasores aproveitam os fluxos de trabalho de sumarização ou tradução para ignorar os controles de segurança e mecanismos de alinhamento.

Uma literatura crescente demonstra que traduzir prompts maliciosos para idiomas com poucos recursos ou menos protegidos, ou reformular solicitações prejudiciais como tarefas de sumarização ou tradução, permite que os adversários evitem as principais medidas de segurança [219, 99]. Estudos empíricos mostram que os LLMs frequentemente falham em filtrar conteúdo perigoso recodificado por meio dessas tarefas, especialmente quando o alinhamento na tradução ou sumarização é fraco [52, 53, 171, 146]. Técnicas como a incorporação de instruções adversárias em modelos de tradução/sumação e a tradução automática com preservação da semântica aumentam ainda mais as taxas de sucesso dos ataques, particularmente em cenários com poucos recursos ou multilíngues. Esses ataques frequentemente escapam da detecção porque imitam solicitações de tarefas legítimas e podem contornar filtros de segurança baseados em palavras-chave e intenção, destacando a necessidade de um alinhamento robusto entre tarefas e multilíngue em estratégias de defesa [168, 125].

III.2 Dependência excessiva de instruções do usuário

Definição: A dependência excessiva de instruções denota uma vulnerabilidade estrutural em LLMs (Modelos de Aprendizado de Máquina) na qual os modelos executam instruções incorporadas em sua entrada de forma excessiva e indiscriminada, independentemente da proveniência, privilégio ou hierarquia contextual. Essa suscetibilidade surge da tendência intrínseca dos modelos de generalizar as capacidades de seguir instruções, carecendo de mecanismos robustos para autenticar ou diferenciar entre comandos confiáveis (sistema/desenvolvedor) e não confiáveis (usuário/ externos), permitindo assim que adversários manipulem o comportamento do modelo por meio de instruções elaboradas.

A literatura recente revela sistematicamente o alcance dessa vulnerabilidade. Hines et al. [70] fornecem uma análise fundamental, demonstrando que os LLMs ajustados para instruções são inerentemente propensos à injeção indireta de prompts, especialmente quando as instruções se originam de dados externos ou controlados pelo usuário; sua abordagem de "destaque" expõe os limites da demarcação de fronteiras de instruções. Kimura et al. [94] mostram empiricamente que os modelos de linguagem de visão obedecem prontamente a instruções adversárias, mesmo quando codificadas como texto em imagens, exacerbado por uma maior conformidade com as instruções. Wallace et al. [181] introduzem uma estrutura de hierarquia de instruções, enfatizando que os LLMs robustos devem priorizar sistematicamente comandos privilegiados em relação aos não privilegiados, com o ajuste fino considerando privilégios resultando em ganhos significativos de robustez. Zhan et al. [229] estendem isso a agentes LLM, onde instruções de ação não filtradas e baseadas em ferramentas de ambientes não confiáveis levam a altas taxas de sucesso de ataque. Estudos sobre LLaMA ajustado para segurança [16] e Banerjee et al. [48] revelam ainda que o aprimoramento do seguimento de instruções, a menos que seja rigorosamente controlado por privilégios, amplifica tanto a utilidade para o usuário quanto a suscetibilidade a prompts maliciosos, causando, por vezes, recusas excessivas ou desvios éticos. Toyer et al. [177] e Chen et al. [24] reforçam que os atacantes exploram a incapacidade dos modelos de distinguir instruções genuínas do sistema de metacomandos adversários, tornando a filtragem de conteúdo ingênua insuficiente. Ren [155] e Nestaas et al. [138] destacam que até mesmo saídas simuladas de agentes ou alegações persuasivas são consideradas confiáveis se formatadas como instruções, evidenciando a inadequação da instrução como autenticação. Coletivamente, esses trabalhos ressaltam que a dependência excessiva de instruções não é meramente uma questão de alinhamento, mas uma falha fundamental enraizada na insuficiente separação de privilégios e na modelagem de confiança, exigindo reformas sistêmicas, arquitetônicas e procedimentais no projeto de LLM.

III.3 Manipulação Psicológica Definição: Manipulação

psicológica refere-se à exploração do alinhamento do LLM por meio de estratégias de incentivo persuasivas ou de engenharia social, aproveitando insights da psicologia, comunicação e ciências sociais.

Em vez de contornar diretamente as proteções por meios técnicos, esta categoria tem como alvo as vulnerabilidades comportamentais do modelo imitando a comunicação persuasiva humana - como apelos emocionais, raciocínio lógico, endossos de autoridade ou enquadramento de cenários - coagindo o modelo a gerar de outra forma

Conteúdo restrito ou prejudicial.

Estudos recentes investigam sistematicamente a ligação entre manipulação psicológica e vulnerabilidades de LLM. Zeng et al. [226] introduzem uma taxonomia abrangente de persuasão, demonstrando que prompts adversários persuasivos (PAPs) construídos com base em princípios das ciências sociais podem consistentemente contornar as medidas de segurança de LLM, alcançando altas taxas de sucesso de jailbreak em modelos de código aberto e fechado. Seus resultados mostram que técnicas como apelo lógico, endosso de autoridade e manipulação emocional são altamente eficazes, especialmente quando adaptadas a domínios de risco específicos. Este trabalho destaca que modelos mais avançados e úteis são paradoxalmente mais suscetíveis a ataques persuasivos semelhantes aos humanos e que as defesas existentes — focadas na detecção de prompts baseados em otimização ou orientados por padrões — são insuficientes para manipulações sutis e ricas em contexto. O Cold-Attack [62] complementa isso, permitindo um controle preciso sobre as características do ataque (por exemplo, sentimento, estilo, fluência contextual), expandindo assim o panorama da manipulação psicológica para incluir cenários de jailbreak mais furtivos e diversos. Essas constatações revelam uma necessidade urgente de conciliar a segurança da IA e as ciências sociais, e de desenvolver defesas adaptativas baseadas na compreensão da suscetibilidade do modelo à influência humana, e não apenas em explorações técnicas.

III.4 Definição de Exploração Hipotética e Baseada em Cenários: A Exploração

Hipotética e Baseada em Cenários denota uma categoria de vulnerabilidades em que os atacantes constroem cenários hipotéticos, ficcionais ou ricamente contextualizados para subverter o alinhamento do Modelo de Aprendizagem Baseado em Lógica (LLM) e os mecanismos de segurança. Essa abordagem manipula as premissas subjacentes do modelo, a lógica situacional e o raciocínio baseado em cenários — frequentemente por meio da encenação de papéis ou da construção narrativa — não para imitar ou personificar um personagem específico, mas para enganar a inferência do modelo sobre o propósito e o contexto do usuário. Diferentemente da Personificação e da Exploração por Interpretação de Papéis (Taxonomia III.8), que se concentram em personificar papéis ou entidades específicas para evocar comportamentos consistentes com o papel, essa classe explora a plausibilidade e a lógica narrativa de situações inteiras, independentemente da identidade assumida pelo usuário.

Estudos recentes revelam que a exploração hipotética e baseada em cenários apresenta riscos distintos, alavancando o enquadramento de cenários com múltiplas interações, o contexto narrativo e a intenção hipotética. Por exemplo, o projeto RED QUEEN demonstra que a inserção de solicitações em cenários plausíveis (por exemplo, ambientes investigativos ou educacionais) contorna sistematicamente os filtros, explorando as limitações dos Modelos de Linguagem Lógica (LLMs) na Teoria da Mente [84]. Os projetos GUARD e Visual-RolePlay mostram ainda que contextos ricos em cenários e gerados dinamicamente podem ser sintetizados em larga escala, inserindo ataques em diálogos aparentemente inofensivos para desafiar a moderação [85, 133]. O projeto A Wolf in Sheep's Clothing generaliza esses ataques, aninhando prompts em contextos narrativos em camadas, desacoplando o ataque de padrões de string estáticos [40]. Trabalhos multimodais, incluindo Image-to-Text Logic Jailbreak e Arondight, estendem esse vetor a cenários visualmente representados, ilustrando que a exploração de cenários não se limita ao texto, mas também afeta modelos de visão-linguagem [252, 121]. Sistemas automatizados e agentes, como SoP, AutoDAN-Turbo e RedAgent, evoluem e otimizam sistematicamente ataques baseados em cenários, aproveitando feedback e memória para descobrir autonomamente novas vulnerabilidades [213, 117, 204]. A pesquisa em engenharia social destaca que cenários baseados em construção de confiança e autoridade amplificam ainda mais o sucesso do ataque, muitas vezes com menos consultas do que solicitações diretas [166].

III.5 Exploração de Aprendizado com Poucos Exemplos e em Contexto A

exploração hipotética e baseada em cenários refere-se a técnicas que utilizam aprendizado com poucos exemplos ou em contexto para contornar medidas de segurança.

III.5.1 Definição de Exploração de Aprendizagem com

Poucos Exemplos: A Exploração de Aprendizagem com Poucos Exemplos refere-se à manipulação de grandes modelos de linguagem (LLMs) por meio da inserção de um pequeno conjunto de pares demonstração-resposta ou exemplos comportamentais cuidadosamente elaborados, diretamente no contexto de estímulo do modelo. Essa abordagem aproveita as capacidades inerentes de aprendizagem contextual e generalização de padrões dos LLMs para induzir comportamentos-alvo.

A literatura recente demonstra que a exploração de poucos disparos representa uma superfície de ataque crítica exclusiva dos LLMs. Trabalhos fundamentais como ICA [197] e adviCL [185] estabelecem que a inserção de um pequeno número de demonstrações prejudiciais ou de recusa no contexto pode aumentar drasticamente o sucesso do ataque, com o número necessário de tentativas crescendo apenas logaritmicamente com a probabilidade de sucesso. Estudos sobre leis de escala [8] revelam que modelos maiores com janelas de contexto mais longas são mais suscetíveis a tais ataques. Análises de transferibilidade [185] mostram que conjuntos de demonstração otimizados podem generalizar para instruções prejudiciais não vistas, e pesquisas recentes [144, 135, 217] estendem essas vulnerabilidades a LLMs multimodais e integrados à visão. Além disso, iterativos ou

Métodos de estímulo contrastivo [36, 81] demonstram que os LLMs podem gerar e refinar autonomamente demonstrações potentes de ataque ou defesa. Em diversos estudos, mecanismos de defesa como filtros de saída, detecção baseada em perplexidade ou estímulos do sistema consistentemente ficam atrás da ameaça adaptativa representada pela exploração com poucos exemplos/em contexto, destacando os riscos fundamentais de alinhamento inerentes a essa categoria à medida que os modelos escalam e as janelas de contexto se expandem.

III.5.2 Definição de Exploração da Cadeia de Pensamento:

A Exploração da Cadeia de Pensamento (CoT) abrange tanto ataques que manipulam a trajetória de raciocínio de grandes modelos de linguagem (LLMs) quanto aqueles que utilizam dicas da CoT como ferramenta para construir estratégias de ataque mais eficazes. Essa classe inclui (1) intervenções adversárias no raciocínio de múltiplas etapas ou composicional do modelo — como inserir elementos de distração, respostas preventivas ou dicas disfarçadas na cadeia de raciocínio para direcionar secretamente as saídas — e (2) metodologias de ataque que exploram explicitamente dicas no estilo CoT (por exemplo, decomposição passo a passo, dramatização ou orientação iterativa) para contornar salvaguardas de alinhamento ou revelar informações sensíveis.

A literatura evidencia ambas as formas de exploração de CoT. Xu et al. [205] demonstram que adversários podem comprometer o raciocínio de LLMs introduzindo dicas de “resposta preventiva” ou elementos de distração antes das cadeias de raciocínio, degradando significativamente a robustez. Bhardwaj et al. [14] mostram que estruturas de “Cadeia de Enunciados” em prompts de múltiplas etapas reduzem as taxas de recusa e permitem conclusões prejudiciais por meio de pensamentos internos manipulados. Jailbreaking em múltiplas etapas [98] e Jailbreaks do tipo "Descubra e Descubra" [109] usam decomposição no estilo CoT, dramatização e adivinhação iterativa para corroer as proteções de privacidade e induzir gradualmente resultados inseguros.

“Inverter” ou disfarçar os prompts [124] manipula o processo de raciocínio, ofuscando conteúdo prejudicial e, em seguida, instruindo o modelo a reconstruí-lo por meio de instruções encadeadas no contexto. No domínio da linguagem visual, os ataques [218] coordenam prompts visuais e textuais do CoT, usando loops de feedback para otimizar estratégias adversárias. Outros trabalhos [48, 139] ilustram que o uso iterativo de prompts com reconhecimento de estrutura, baseado em princípios do CoT, amplifica resultados antiéticos. Finalmente, pipelines composicionais [89] aproveitam a decomposição de tarefas no estilo CoT em vários modelos, encadeando subtarefas benignas para alcançar resultados gerais inseguros. Abordagens de cadeia de ataque com múltiplas rodadas [212] ilustram como prompts graduais e contextualmente vinculados podem guiar progressivamente os modelos à conformidade.

III.6 Definição de Exploração de Ambiguidade Contextual: A

Exploração de Ambiguidade Contextual refere-se a técnicas adversárias que aproveitam incertezas ou ambiguidades no contexto de uma entrada para subverter as fronteiras semânticas que os grandes modelos de linguagem (LLMs) utilizam para a execução de tarefas e moderação de conteúdo. Ao contrário dos ataques de prompt baseados em manipulação explícita ou gatilhos em nível de token, esta categoria explora confusões sutis ou ardilosas, obscurecendo as pistas contextuais que permitem aos LLMs distinguir entre intenções benignas e maliciosas.

A literatura recente demonstra consistentemente que os LLMs são vulneráveis à ambiguidade contextual. Shang et al. [132] mostram que consultas ambíguas ou ofuscadas podem induzir confusão e evitar a detecção de intenções maliciosas, mesmo em modelos robustos. Mei et al. [139] destacam que conflitos e ambiguidade contextual são fontes importantes de falsos positivos/negativos na detecção de jailbreak, enfatizando a necessidade de avaliadores sensíveis à incoerência contextual. Ren et al. [154] revelam que ataques de múltiplas etapas podem alterar sutilmente a semântica em contextos aparentemente benignos, explorando a dependência dos LLMs na continuidade contextual. Bianchi et al. [16] relatam que LLMs ajustados para segurança podem confundir entradas ambíguas, mas seguras, com conteúdo inseguro devido à capacidade limitada de desambiguação. Pasquini et al. [141] demonstram que limites inadequados entre instruções e dados permitem que payloads adversários sejam classificados erroneamente como contexto benigno. Pellegrino et al. [169] fornecem evidências de que os LLMs baseados em RAG são particularmente suscetíveis à ambiguidade de contexto, uma vez que o conteúdo recuperado ambíguo ou contraditório pode facilitar a manipulação indireta do prompt. Juntos, esses trabalhos ressaltam que a exploração da ambiguidade de contexto continua sendo uma vulnerabilidade crítica e insidiosa, exigindo métodos aprimorados para o desvendamento do contexto e a inferência de intenção para garantir a segurança do LLM.

III.7 Definição de Exploração de Conformidade Condicional: A

Exploração de Conformidade Condicional refere-se a uma subcategoria sutil de vulnerabilidades de LLM (Modelagem de Aprendizado de Liderança) na qual os adversários obtêm resultados inseguros ou que violam as políticas não por meio de instruções explícitas, mas manipulando as condições sob as quais os modelos julgam a conformidade. Essa exploração depende do contexto, da estrutura da instrução ou do estado do modelo — desencadeando comportamentos prejudiciais somente quando pistas semânticas, sintáticas ou posicionais específicas estão presentes, e permanecendo benigna caso contrário.

A literatura recente revela sistematicamente os mecanismos e o alcance da exploração da conformidade condicional. Ataques universais de backdoor [150, 96] demonstram que gatilhos secretos ou sufixos adversários podem subverter universalmente as recusas, preservando as saídas normais na sua ausência, o que destaca os desafios de detecção. DeepInception [105] mostra que a incorporação de solicitações prejudiciais em cenários aninhados, baseados em autoridade ou orientados por papéis, explora vulnerabilidades psicológicas, permitindo a geração de conteúdo prejudicial sem entrada explícita de conteúdo inseguro. Em ambientes multimodais, gatilhos baseados em estrutura, como a codificação visual de texto prejudicial [193] ou arranjos específicos de prompts [51], expandem ainda mais a conformidade condicional para novas modalidades. Estudos sobre edição de modelos [69] expõem como os limites condicionais são definidos por artefatos de treinamento e edição refinados, com a segurança deixando caminhos de conformidade abertos ou induzindo recusas exageradas. Manipulação social, psicologia reversa e técnicas de troca [63] ilustram a facilidade prática de criar estados dependentes do contexto que geram conteúdo proibido. Notavelmente, um trabalho recente [183] expõe a profundidade estrutural dessa vulnerabilidade, mostrando que caminhos de decodificação baseados em custos podem revelar conclusões prejudiciais ao longo de sequências alternativas, acionadas condicionalmente. Coletivamente, esses estudos ressaltam que a exploração da conformidade condicional é adaptativa, muitas vezes invisível para testes superficiais e difícil de mitigar sem alinhamento sensível ao contexto, destacando uma compensação persistente entre a capacidade do modelo e a segurança.

III.8 Definição de Personificação e Exploração de Papéis: A

exploração de personificação e interpretação de papéis é uma classe de exploração de vulnerabilidades do LLM caracterizada pela manipulação das faculdades de personificação e imaginação ficcional do modelo por meio de instruções baseadas em personagens. Esse método subverte as salvaguardas de segurança, induzindo o modelo a adotar papéis ou a se envolver em narrativas imaginadas, levando-o a suspender temporariamente os julgamentos normativos de segurança. Ao contrário dos ataques em nível de token ou baseados em codificação, a exploração de papéis explora o design conversacional e de seguimento de instruções do LLM, aproveitando a adoção de personas para mascarar, distrair ou justificar conteúdo malicioso dentro de um contexto ficcional.

A literatura recente converge para o princípio de que a encenação de papéis induzida sistematicamente possibilita violações normativas que de outra forma seriam bloqueadas. Yang et al. [213] introduzem o SEQAR, que gera e otimiza automaticamente múltiplos personagens de fuga da prisão para maximizar a distração e a divisão de contexto, cada um respondendo sequencialmente como agentes distintos. O DeepInception de Li et al. [105] enquadra a encenação de papéis como uma inception psicológica, empilhando cenas ficcionais aninhadas e hierarquias de personagens de autoridade-submissão para induzir a “auto-perda” no modelo e amplificar a saída prejudicial.

III.9 Exploração de Recursos e Limitações do Sistema A exploração

de recursos e limitações do sistema refere-se a ataques que se aproveitam de propriedades intrínsecas de sistemas LLM — como abuso de chamadas de função, vazamento de prompts do sistema e manipulação do comprimento do contexto. Esses ataques revelam que mesmo modelos bem alinhados podem ser comprometidos por meio de fragilidades no projeto em nível de sistema, enfatizando a necessidade de mecanismos mais robustos de isolamento, validação de entrada e integridade contextual.

III.9.1 Definição de Exploração de Chamada de

Função: A Exploração de Chamada de Função é uma subclasse da Exploração de Recursos do Sistema que visa especificamente a capacidade de chamada de função em grandes modelos de linguagem (LLMs). Essa vulnerabilidade surge quando adversários manipulam a interface de chamada de função do sistema — destinada ao uso de ferramentas ou à integração de APIs — para forçar caminhos de execução que contornam os controles de alinhamento e segurança, frequentemente explorando validação de entrada insuficiente ou esquemas de API excessivamente permissivos.

A literatura demonstra a potência e a generalidade da exploração de chamadas de função como vetor de ataque. Wu et al. [170] fornecem um estudo sistemático, revelando que os atacantes podem criar entradas de chamadas de função que evitam o alinhamento em nível de prompt e em nível de modelo, levando a jailbreaks confiáveis em vários modelos e configurações operacionais. Seus resultados destacam que as estratégias padrão de filtragem e alinhamento são inadequadas quando as interfaces de nível de sistema, como chamadas de função, não são suficientemente reforçadas, expondo uma superfície de ataque profunda e transferível que não pode ser abordada apenas por meio de engenharia de prompt ou curadoria superficial de conjuntos de dados.

III.9.2 Definição de Vazamento de Prompt

do Sistema: Vazamento (Roubo) de Prompt do Sistema é uma classe de vulnerabilidade na qual adversários extraem ou manipulam o prompt confidencial do sistema ou instruções ocultas de um modelo, explorando características inerentes ou operacionais de sistemas baseados em LLM. Ao contrário da injeção de prompt convencional, essa classe visa contextos privilegiados e não expostos ao usuário, permitindo ataques subsequentes — como roubo de prompt, desbloqueio persistente ou extração de políticas comportamentais sensíveis — sem exigir acesso aos detalhes internos do modelo.

Estudos recentes revelam que os invasores podem exfiltrar sistematicamente os prompts do sistema aproveitando o LLM

mecanismos de tratamento de contexto e seguimento de instruções. Wu et al. [200] demonstram o roubo de prompts em LLMs multimodais por meio da manipulação de diálogos baseada em API, automatizando inclusive a conversão de prompts roubados em payloads de jailbreak. Geiping et al. [56] ampliam isso categorizando ataques de “extração de prompts” e “repetição de prompts”, mostrando que a entrada adversária pode atuar como “código arbitrário” para extrair diretivas ocultas sob controles rígidos de interface do usuário. Liu et al. [122] destacam os riscos na fronteira código-dados em aplicações LLM, onde falhas na separação de contexto facilitam o vazamento. No lado da defesa, Khomsky et al. [91] constatarem que proteções multicamadas permanecem vulneráveis à evasão por meio de saídas ofuscadas, enquanto Hines et al. [70] propõem o “destaque” — o uso da codificação de proveniência de prompts para reduzir a extração. Em conjunto, esses trabalhos demonstram que o vazamento de prompts do sistema explora vulnerabilidades únicas no limite do contexto privilegiado, muitas vezes contornando a sanitização padrão e destacando a necessidade de mecanismos de isolamento mais robustos na segurança de LLM (Loading and Management Layer).

III.9.3 Definição de Exploração do Comprimento do

Contexto: A Exploração do Comprimento do Contexto refere-se a técnicas adversárias que aproveitam a janela de contexto baseada em tokens e suas limitações de comprimento para manipular grandes modelos de linguagem (LLMs) e produzir resultados não intencionais ou inseguros.

A literatura recente investiga sistematicamente a exploração do comprimento do contexto. Schulhoff et al. [158] identificam ataques como o “Context Overflow”, em que os adversários preenchem prompts para deslocar ou suprimir instruções benignas, manipulando o comportamento do modelo por meio do esgotamento do orçamento de tokens. Anil et al. [8] generalizam isso por meio do “Many-Shot Jailbreaking”, demonstrando que o aumento do comprimento do contexto permite que os atacantes ignorem de forma confiável o alinhamento de segurança, preenchendo o contexto com demonstrações adversárias; eles mostram que a eficácia do ataque aumenta com o tamanho da janela e não é totalmente mitigada pelo alinhamento padrão. Geiping et al. [56] expandem ainda mais esse escopo, mostrando que a manipulação do contexto permite a extração de prompts, o redirecionamento de saída e a exploração de peculiaridades internas do modelo, como “tokens de falha”. Esses trabalhos, em conjunto, mostram que a exploração do comprimento do contexto é uma vulnerabilidade fundamental enraizada na arquitetura LLM, desafiando a integridade e a segurança dos LLMs à medida que as janelas de contexto aumentam.

3 Conjunto de dados

Durante nosso levantamento de estudos sobre ataques de jailbreak, defesas contra jailbreak e pesquisas relacionadas, coletamos e organizamos os conjuntos de dados disponibilizados por diversas fontes. Como resultado, compilamos o maior conjunto de dados disponível publicamente sobre ataques de jailbreak na comunidade de código aberto. Além disso, compilamos um conjunto de dados com mais de um milhão de prompts benignos.

Para investigar o panorama atual de conjuntos de dados de código aberto relacionados a ataques de jailbreak em LLMs (Lower Liability Management), realizamos um levantamento abrangente da literatura relevante. Nossa coleção inclui 31 trabalhos relacionados a visão geral e benchmarks de jailbreak, 84 trabalhos sobre ataques de jailbreak de caixa-preta e 15 trabalhos sobre ataques de jailbreak de caixa-branca. Coletamos e organizamos os conjuntos de dados de código aberto de todos os trabalhos para formar o maior conjunto de dados de jailbreak e prompts benignos da comunidade até o momento, contendo 445.752 dados de prompts de jailbreak de 48 fontes e 1.094.122 dados de prompts benignos de 14 fontes.

O conjunto de dados está organizado em cinco colunas: prompt do sistema, prompt do usuário, jailbreak, source e tactic. _____ As colunas “system prompt” e “user prompt” denotam, respectivamente, o prompt do sistema e o prompt do usuário submetidos ao modelo. A coluna “jailbreak” indica se o prompt constitui uma tentativa de jailbreak (1 para jailbreak, 0 para benigno). A coluna “source” indica a origem dos dados e a coluna “tactic” indica se uma tática de jailbreak foi empregada (1 para prompts que utilizam táticas de jailbreak, 0 para prompts que não utilizam táticas de jailbreak). As Tabelas 2 e 3 apresentam, respectivamente, estatísticas sobre o número de dados e o comprimento médio dos prompts para cada origem nos conjuntos de dados de prompts de jailbreak e prompts benignos.

Tabela 2: Informações do conjunto de dados de prompts de jailbreak

Fonte	Contagem de duração média do prompt	
allenai/wildjailbreak [82] 134778		648,28
ReNeLLM [40] 125494		548,49
FuzzLLM [215] 62136		1758,48
AetherPrior/TrickLLM [151] 40245		6084,99
GPTFuzzer [220] 11808		2088.01
CPAD [111] 10050 yueliu1999/FlipGuardData [124] 8840		127,38
		974,66
Conhecimento para quebrar a prisão [178] 7712		909,14
SoftMINER-Group/TechHazardQA [13] 7301 h4rm3l [44] 5294 sufixo-talvez-recurso/		96,60
sufixo-talvez-recursos-de-anúncio [239] 4556		852.01
		78,08
TemplateJailbreak [208] 4100		2222,62
ECLIPSE [83] 4021		109,67
SAP [36] 2120		794,73
AdaPPA [130] 1948 desbloqueado [196]		436,19
	1898	963,47
Safe-RLHF [149] tml-	1794	68,39
epfl/llm-passado [6]	1419	101,08
E [163]	1398	2681,40
ACE [65]	1050	959,77
TAP [137]	831	508,37
JUIZ DA PRISÃO [112]	826	357,18
Gerar GPT [128]	763	72,31
PAR [22]	741	464,19
Ataque adaptativo [4]	689	1805,27
AdvBench [251]	515	73,01
Caudas de castor [77]	493	70,56
Fuga de prisão WUSTL-CSPL/LLM [222]	447	1756,81
HarmBench [136]	411	88,08
StrongREJECT [168]	238	168,28
Conjunto de perguntas [128]	215	74,44
CARREGAR [115]	212	1496,65
GCG [251]	200	223,45
AutoDAN [248] hh-	187	3416,41
rlhf [128]	167	83,86
PAP [226]	165	1029,44
GBDA [61]	137	582,16
Artesanato [128]	124	107,77
Instrução Maliciosa [74]	100	61,70
Reescrita GPT [128]	96	66,69
DeepInception [105]	64	558,66
Comportamentos JBB [21]	55	95,42
ConjuntoGCG [251]	34	613,15
UAT [180]	33	401,73
AutoPrompt [165]	21	741,86
DirectRequest [136]	15	1198,47
Estudo LLM sobre fugas da prisão [128]	8	129,00
MasterKey [37]	3	77,67

Tabela 3: Informações do conjunto de dados de estímulo benigno

Fonte	Contagem de duração média do prompt	
OpenHermes-2.5 [173] glaive-	494829	927,15
code-assist [1] allenai/	181505	409,57
wildjailbreak [82]	128963	520,90
CamelAI [18] 77276 EvolInstruct 70k [199] 44393		218,21
cot alpaca gpt4 [33] 42022 metamath [221] 36596		559,97
airoboros2.2 [87] 35320 platypus [97] 22126 DAN		85,96
[163] 16989 UnnaturalInstructions [143] 6595		244,34
CogStackMed [29] 4408 LMSys Chatbot Arena		487,79
[243] 3000 JBB-Behaviors [21]		547,99
		1552,36
		369,20
		211,15
		192,38
	100	73,75

4 PromptSecurity: Uma Plataforma Modular Unificada para LLM Prompt Avaliação de Segurança

4.1 Visão geral

Este capítulo apresenta a plataforma PromptSecurity — uma estrutura modular, sistemática e flexível para avaliação ponta a ponta de grandes modelos de linguagem. Abrange diversas configurações de ataque, defesa e modelo, permitindo comparações justas e abrangentes entre métodos arbitrários sob um protocolo de avaliação unificado.

Motivação e Limitações Estudos anteriores sobre avaliação rápida de segurança permanecem fragmentados, frequentemente carecendo de um modelo de ameaça unificado e de um protocolo de avaliação padronizado. Isso leva a resultados inconsistentes e potencialmente prejudiciais. comparações injustas, como quando alguns ataques são auxiliados por LLMs auxiliares mais fortes do que outros. Além disso, A ausência de uma arquitetura modular e interoperável limita a flexibilidade e a escalabilidade à medida que novos métodos surgem. O PromptSecurity aborda esses desafios por meio de uma estrutura de avaliação unificada e extensível.

4.2 Objetivos e pressupostos do projeto

O design do PromptSecurity é guiado por três princípios abrangentes que abordam as limitações estruturais identificadas em esforços de avaliação anteriores.

- **Arquitetura de Avaliação Modular.** O PromptSecurity abstrai o processo de avaliação em um conjunto de Componentes fundamentais e ortogonais — Ataques, Defesas, Modelos, Avaliadores e Conjuntos de Dados. Cada um. O componente adere a uma especificação de interface unificada, permitindo composição arbitrária entre métodos heterogêneos e garantindo que qualquer ataque ou defesa possa ser avaliado sob condições experimentais consistentes. suposições. Essa modularidade estabelece uma base de princípios para uma comparação sistemática e justa.
- **Gestão de Experimentos Unificada e Escalável.** Para garantir a reprodutibilidade e a extensibilidade, o Prompt-Security utiliza um sistema de gestão orientado a configurações, onde todos os comportamentos experimentais são especificados externamente por meio de arquivos de configuração declarativos (por exemplo, JSON/YAML). Essa separação entre A configuração e implementação facilitam a experimentação escalável: novos métodos podem ser incorporados sem problemas. integrado e avaliado em lote nos módulos existentes, preservando a reprodutibilidade e a versão rastreabilidade.
- **Usabilidade e Manutenibilidade por Design.** A estrutura enfatiza ainda mais a facilidade de extensão. e capacidade de manutenção a longo prazo. Através da descoberta automática de componentes, validação de configuração e

Graças aos mecanismos de repetição determinísticos, o PromptSecurity minimiza a intervenção humana e reduz a barreira de engenharia para a integração de novos ataques, defesas e conjuntos de dados. Essa filosofia de design garante a sustentabilidade da estrutura à medida que o ecossistema de métodos de segurança imediata evolui rapidamente.

4.3 Módulos de Componentes

4.3.1 Módulo de Ataque

O módulo de ataque operacionaliza a geração de jailbreak em nossa estrutura sob um modelo de ameaça claro: um adversário manipula entradas visíveis ao usuário no LLM alvo sem alterar os pesos do sistema ou a infraestrutura. Consideramos tanto o acesso de caixa-preta (somente consultas e observações de saídas) quanto o acesso de caixa-branca (pesos/gradientes/logits disponíveis) e permitimos que o alvo seja qualquer backend em conformidade com o BaseModel, incluindo APIs comerciais, modelos locais do HuggingFace ou wrappers protegidos (BaseDefendedModel). Esse design garante que os ataques se integrem naturalmente às defesas em avaliações de ponta a ponta e sejam comparáveis entre diferentes modalidades de implantação.

Cada implementação de ataque herda de BaseAttack e segue um procedimento padronizado que, dada uma descrição benigna da tarefa e configuração opcional, produz (i) um orçamento cumulativo de consultas — o número total de chamadas de modelo despendidas durante a construção do prompt — e (ii) um candidato a prompt adversário a ser emitido para o alvo. A contabilização de consultas é holística: se um ataque utiliza componentes auxiliares (por exemplo, um LLM de atacante/reescritor/avaliador/julgador), suas invocações são incluídas no mesmo orçamento. O módulo é orientado por configuração e reproduzível: os métodos são descobertos dinamicamente, os parâmetros são especificados em JSON e os portões de capacidade restringem os algoritmos de caixa branca a backends acessíveis por peso.

Para permitir uma análise de cobertura baseada em princípios, em vez de agrupamentos específicos de implementação, anotamos cada método com uma ou mais etiquetas da Taxonomia I (Figura 1) — por exemplo, “1.1.1 Ofuscação e Codificação” ou “1.2.3 Aprendizado por Reforço”. A versão atual inclui 18 técnicas (15 de caixa-preta e 3 de caixa-branca), além de uma linha de base sem ataques; um resumo das suposições de acesso, uso de LLM auxiliar e rótulos da taxonomia aparece na Tabela reftab:attack-module.

Em consonância com a Taxonomia I (2.3), anotamos cada implementação com um ou mais rótulos de taxonomia (por exemplo, “1.1.1 Ofuscação e Codificação”, “1.2.3 Aprendizagem por Reforço”) para permitir análises de cobertura fundamentadas e ablações estratificadas.

4.3.2 Módulo de Defesa

O módulo de defesa fornece um wrapper controlado pelo defensor que transforma qualquer backend compatível com BaseModel em um LLM protegido. O wrapper intermedia todas as interações: todas as entradas para o sistema — incluindo consultas geradas por ataques e prompts de múltiplas etapas — são aplicadas ao modelo protegido, e não ao modelo bruto. As intervenções abrangem três níveis: entrada (sanitização e reescrita antes da geração), modelo (roteamento de políticas, diagnósticos de estado oculto/gradiente, reforço baseado em otimização) e saída (filtragem pós-geração e autoavaliação). Cada defesa cria uma subclasse de BaseDefendedModel e executa um pipeline de defesa padronizado (se houver), ou seja, defesa de entrada, defesa de modelo e defesa de saída, com passagem de recursos e controle automático (por exemplo, métodos baseados em gradiente exigem backends locais de precisão total com acesso a pesos). As defesas são orientadas por configuração, com valores padrão por método; verificações de compatibilidade impedem configurações inválidas. Vários mecanismos de defesa podem ser aplicados ao LLM alvo de forma em cascata ou paralela. A Tabela 5 resume o local de intervenção e o uso de LLM auxiliar. Para uma análise baseada em princípios, as defesas são classificadas usando a Taxonomia II (Figura 2).

Tabela 4: Ataques de jailbreak implementados, suposições de acesso, uso de LLMs auxiliares e tags de taxonomia. LLMs auxiliares referem-se a componentes modulares de atacante/reescritor/avaliador/julgador carregados por meio da estrutura.

Método	Acesso	Etiquetas de taxonomia de LLMs auxiliares	
nenhum ataque	Linha de base	Não	Linha de base (sem perturbação)
FlipAttack [124]	Caixa preta n°		1.1.1 Ofuscação e Codificação
ArtPromptAttack [80]	Caixa preta Sim		1.5.1 Aproveitando as vulnerabilidades comportamentais
PAR [23]	Caixa preta Sim		1.2.1 Algoritmos Heurísticos; 1.3.2 Colaboração Multiagente; 1.3.3 Modelos Substitutos; 1.4.2 Escalada Gradual
ABJAttack [109]	Caixa preta Não		1.1.4 Ocultação de Intenção; 1.5.4 Bypass de Mecanismos de Defesa
Ataque no Passado [7]	Caixa preta Sim		1.1.2 Substituição e Sinônimos
Ataque IFSJAttack [244]	Caixa preta Não		1.2.2 Estimativa de Gradiente
Ataque de código [153]	Caixa preta Não		1.1.1 Ofuscação e Codificação
Código Camaleão [129]	Caixa preta Não		1.1.1 Ofuscação e Codificação
CARREGAR [115]	Caixa preta n°		1.5.1 Aproveitando as vulnerabilidades comportamentais
TapAttack [137]	Caixa preta Sim		1.2.1 Algoritmos Heurísticos; 1.3.2 Colaboração Multiagente; 1.4.1 Ataques contextuais de múltiplos turnos
DrAttack [104]	Caixa preta Não		1.4.1 Ataques contextuais de múltiplos turnos
Ataque de Inception [105]	Caixa preta Não		1.1.4 Ocultação de Intenção; 1.5.4 Bypass de Mecanismos de Defesa
ReNeLLM [40]	Caixa preta Sim		1.2.1 Algoritmos Heurísticos; 1.3.1 Ataques Gerados por LLM; 1.4.2 Escalada Gradual
GPTFUZZER [220]	Caixa preta Sim		1.2.3 Aprendizado por Reforço
Ataque PersuasivoNoContexto [226]	Caixa-preta Sim		1.1.4 Ocultação de Intenção; 1.3.1 Ataques Gerados por LLM; 1.5.1 Aproveitando as vulnerabilidades comportamentais
GCGWhiteBoxAttack [251]	Caixa branca Não		2.2.1 Manipulação de Prefixo/Sufixo; 2.2.2 Otimização Baseada em Gradiente
AutoDANAttack [117]	Caixa branca Não		2.2.1 Manipulação de Prefixo/Sufixo
Ataque COLDAck [62]	Caixa branca n°		2.2.2 Otimização Baseada em Gradiente

Tabela 5: Defesas implementadas por locus, uso de LLM auxiliar e etiquetas da Taxonomia II (Figura 2). "Auxiliar LLMs" indica o uso opcional de componentes de atacante/reescritor/avaliador/julgador.

Método	Locus (Entrada/Modelo/Saída)	Etiquetas auxiliares de LLMs Taxonomia II	
Sem Defesa	Linha de base	Não	Linha de base (sem intervenção)
Defesa do Filtro de Entrada	Entrada	Sim	1.1.1 Análise de Prompt; 2.1.1 Modificação de Prompt
Defesa OutputFilter	Saída	Sim	2.4.1 Filtragem de saída; 2.4.2 Autoavaliação
Defesa de Perplexidade [2]	Saída	Não	1.2.2 Análise de Probabilidade
Defesa de retrotradução [192]	Entrada	Sim	1.1.2 Detecção de Intenção; 2.1.1 Modificação de Solicitação
Defesa SmoothLLM [156]	Modelo	Sim	2.1.1 Modificação imediata; 2.4.2 Autoavaliação
Defesa do Guarda Prisional [232]	Modelo	Não	1.2.1 Detecção Semântica
PrimeGuard Defense [134]	Modelo	Não	2.1.2 Avisos de segurança
GradSafe Defense [202]	Modelo	Não	1.3.2 Análise de Gradiente
Entrada RobustOpt Defense (RPO) [245]		Não	2.1.2 Avisos de segurança

4.3.3 Módulo do Modelo

O módulo de modelo fornece uma interface unificada e orientada a configuração para APIs hospedadas e modelos servidos localmente. O objetivo é simplicidade e consistência: a troca de fornecedores requer apenas uma alteração de configuração, permitindo Experimentos de ataque e defesa em lote de grande escala, sem necessidade de edição de código. Todos os modelos são invocados por meio de configurações simplificadas. com parâmetros harmonizados para que os experimentos permaneçam comparáveis entre os provedores. Em nossa referência Na implementação, estão disponíveis 73 configurações de API e 53 configurações locais. Como as defesas atuam como wrappers, qualquer modelo carregado pode ser transformado em um LLM protegido; todas as entradas (incluindo geradas por ataque As consultas são aplicadas ao modelo protegido em vez do backend bruto.

A semântica de acesso é definida por categoria: os backends de API são tratados como caixa-preta; os backends locais são Comporta-se como uma caixa branca quando a precisão o permite (por exemplo, pesos de precisão total) e, caso contrário, como uma caixa preta. As tabelas a seguir resumem as configurações padrão, agrupadas por API e famílias locais, com marca/provedor e tamanhos nominais consolidados por família.

Tabela 6: Configurações do modelo de API incluídas no PromptSecurity (nome do modelo, marca/provedor, tamanho nominal).
Todos os backends da API são tratados como caixas-pretas.

Modelo (nome da configuração)	Marca/Empresa	Provedor de servidor	Tamanho
01-ai-Yi-34B-Chat	01.IA	DeepInfra	34B
Autismo-chronos-hermes-13b-v2	–	DeepInfra	13B
Gryphe-MythoMax-L2-13b Gryphe-	Grifo	DeepInfra	13B
MythoMax-L2-13b-turbo HuggingFaceH4-zephyr-	Grifo	DeepInfra	13B
orpo-141b-A35b-v0.1 NousResearch-Hermes-3-Llama-3.1-405B	AbraçandoFaceH4	DeepInfra	141B
	Meta	DeepInfra	405B
Pré-visualização Qwen-QwQ-32B	Alibaba	DeepInfra	32B
Qwen-Qwen2-Instruir Qwen-	Alibaba	DeepInfra	7B, 72B
Qwen2.5-Instruir Qwen-	Alibaba	DeepInfra	7B, 32B, 72B
Qwen2.5-Coder-Instruir	Alibaba	DeepInfra	32B
Sao10K-L3-70B-Euryale-v2.1	Sao10K	Antrópico	70B
	Sao10K	Antrópico	70B
claude-2.0	Antrópico	Antrópico	–
claude-2.1	Antrópico	Antrópico	–
claude-3-5-haicaï-20241022	Antrópico	Antrópico	–
claude-3-5-soneto	Antrópico	Antrópico	— (versões: 20240620, 20241022, mais recente)
claude-3-haicaï-20240307	Antrópico	Antrópico	–
claude-3-opus	Antrópico	Antrópico	— (versões: 20240229, mais recente)
claude-3-sonnet-20240229	Cognitivecomputations-	dolphin-2.6-mixtral-8x7b MistralAI	–
Claude-4	DeepInfra	–	–
claude-instant-1.2	Cognitivecomputations-dolphin-2.9.1-	–	–
claude-soneto-4-20250514	llama-3-70b CognitiveComputations	–	–
deepseek-ai-DeepSeek-R1-0528-Turbo DeepSeek deepseek-ai-DeepSeek-V2.5 DeepSeek DeepSeek deepseek-ai-	DeepInfra DeepInfra deepinfra-airoboros-70b	–	7B
DeepSeek-V3 DeepSeek deepseek-ai-deepseek-chat deepseek-ai-deepseek-coder DeepSeek ByteDance (Ark) 32K	–	–	70B
	DeepInfra	–	70B
	Busca Profunda	–	–
	Busca Profunda	–	–
	Busca Profunda	–	–
	Busca Profunda	–	–
	Busca Profunda	–	–
doubao-1-5-pro-32k-250115	ByteDance	–	–
doubao-seed-1-6-250615 ByteDance (Folha) —	ByteDance	–	–
doubao-seed-1-6-flash-250615	ByteDance	ByteDance (Ark) —	–
gemini-1.0-pro	Google	Google	— (versões: base, mais recente)
gemini-1.5-flash	Google	Google	— (versões: base, 002, mais recente)
gemini-1.5-pro	Google	Google	— (versões: base, 002, mais recente)
gemini-2.0-flash	Google	Google	— (versões: base, exp, thinking-exp-1219)
gemini-2.5-flash	Google	Google	–
gemini-2.5-pro	Google	Google	–
google-gemma-1.1-7b-it google-	Google	DeepInfra	7B
gemma-2-it gpt-3.5-turbo	Google	DeepInfra	7B, 9B, 27B
gpt-4 gpt-4.1 gpt-4o	OpenAI	OpenAI	— (versões: base, 0125, 1106)
	OpenAI	OpenAI	— (versões: base, turbo)
	OpenAI	OpenAI	— (versões: base, mini, nano)
	OpenAI	OpenAI	— (versões: base, mais recente, mini)
o1	OpenAI	OpenAI	–
o1-mini	OpenAI	OpenAI	–
o1-visualização	OpenAI	OpenAI	–
gpt-o3	OpenAI	OpenAI	–
meta-llama-llama-2-chat-hf	Meta	DeepInfra	7B, 13B, 70B
meta-llama-llama-3.2-Vision-Instruct	Meta	–	11B, 90B
meta-llama-llama-3.3-70B-Instruir	Meta	–	70B
meta-llama-llama-4-405B-Instruir	Meta	–	405B
meta-llama-Meta-llama-3.1-Instruir	Meta	–	8B, 70B, 405B
microsoft-Phi-3-médio-4k-instrução	Microsoft	–	–
microsoft-WizardLM-2-7B	Microsoft	–	7B
microsoft-WizardLM-2-8x22B	Microsoft	–	22B
mistralai-Mistral-7B-Instruct-v0.3	MistralAI	–	7B
mistralai-Mistral-Grande-Instrução-2407	MistralAI	–	–
mistralai-Mixtral-8x22B-Instruct-v0.1	MistralAI	–	22B
nvidia-llama-3.1-Nemotron-70B-Instruções	Meta	–	70B
nvidia-Nemotron-4-340B-Instruções	NVIDIA	–	340B
openchat-openchat-3.6-8b	OpenChat	–	8B

Tabela 7: Configurações de modelo local incluídas no PromptSecurity (nome do modelo, marca/fornecedor, nominal tamanho). "Caixa branca*" indica acesso em nível de gradiente quando a precisão o permite.

Modelo (nome da configuração)	Provedor de servidor de marca/empresa		Tamanho
01-AI-Yi-1.5-Bate-papo	01.IA	6B, 9B, 34B	
Qwen-QwQ-32B-Prévia	Alibaba	32B	
Qwen-Qwen2-Instruir	Alibaba	0,5 bilhões, 1,5 bilhões, 7 bilhões, 72 bilhões	
Qwen-Qwen2.5-Instruir	Alibaba	0,5B, 1,5B, 3B, 7B, 14B, 32B, 72B	
Qwen-Qwen2.5-Codificador-Instruções	Alibaba	1,5 bilhões, 7 bilhões, 32 bilhões	
Qwen-Qwen3	Alibaba	0,6 bilhões, 1,7 bilhões, 4 bilhões, 8 bilhões, 14 bilhões, 32 bilhões	
google-gemma-2-it google-gemma-3-1b-it internlm-internlm2-5-1.8b-chat	Google	2B, 9B, 27B	
internlm-internlm2-5-7b-chat	Laboratório de IA de Xangai 1,8B	1B	
internlm-internlm2-5-20b-chat	Laboratório de IA de Xangai 7B		
meta-lhama-Llama-2-chat-hf	Laboratório de IA de Xangai 20B		
meta-lhama-Llama-3-70B-Instruir	Meta 7B, 13B, 70B		
meta-lhama-Llama-3.1-Instruir	Meta	8B, 70B	
meta-lhama-Llama-3.2-Vision-Instruct	Meta	8B, 70B, 405B (incluindo FP8)	
meta-lhama-Llama-3.3-70B-Instruir	Meta	11B, 90B	
meta-lhama-Llama-4-Escoteiro-17B-16E-Instruir	Meta	70B	
microsoft-Phi-2-instruct		17B	
microsoft-Phi-3-medium-4k/128k-instruct	Microsoft	—	
mini-4k/128k-instruct	Microsoft	—	
microsoft-Phi-3-small-4k/8k/128k-instruct	Microsoft	—	
microsoft-Phi-3.5-MoE-instruct	Microsoft	—	
mini-instruções	Microsoft	—	
microsoft-Phi-4-instruct	Microsoft	—	
mistralai-Ministral-8B-Instruct-2410	MistralAI	8B	
mistralai-Mistral-7B-Instruct-v0.3	MistralAI	7B	
mistralai-Mistral-Nemo-Instruct-2407	MistralAI	—	

Nota: Os backends da API são tratados como caixa-preta. "Caixa-branca*" indica que o acesso em nível de gradiente está disponível para backends locais quando a precisão escolhida o permite (por exemplo, pesos de precisão total). Isso é controlado por configuração. A abstração permite a troca entre provedores com um único clique e suporta avaliações de ataque e defesa em lote de grande porte sem alterações no código experimental. Vale ressaltar que todas as APIs e LLMs locais podem servir como os LLMs assistidos em ataques, defesas e julgamentos quando o acesso é compatível.

4.3.4 Módulo Judger

O módulo Judger padroniza a semântica para avaliar a nocividade e o sucesso do ataque. Ele oferece uma Interface consistente para avaliação de instância única e em lote, e compreende três famílias de implementação: (1) juízes baseados em regras (determinísticos, sem modelo), (2) juízes de modelo local (caixa branca quando viável) e (3) Avaliadores de modelo de API (caixa preta). Essa separação preserva a comparabilidade entre experimentos, ao mesmo tempo que permite concessões deliberadas entre precisão, latência e custo.

Semântica e resultados. Por padrão, os avaliadores retornam decisões binárias: 1 indica prejudicial ou ataque bem-sucedido, e 0 denota segurança ou falha. Certos modelos produzem primeiro pontuações graduadas (por exemplo, uma escala de política de 1 a 5) que são mapeado deterministicamente para um resultado binário para manter a comparabilidade entre os avaliadores. Avaliação em lote. é suportado para taxa de transferência, com tratamento uniforme de cotas, limites de taxa e erros de rede por meio de estrutura. exceções.

Backends e intercambiabilidade. Os juízes locais são executados em backends Hugging Face (CPU/GPU), permitindo Avaliação offline e determinismo forte. Os avaliadores de API são independentes do provedor: substituindo o modelo de avaliação. A configuração é suficiente para alternar para qualquer modelo de API compatível (por exemplo, OpenAI, Anthropic, Google, DeepSeek, etc.). DeepInfra) sem alterações de código. Esse design permite comparações diretas entre diferentes provedores.

Modelos e calibração. Os avaliadores de API suportam vários modelos de prompt (por exemplo, nocividade binária, Estilo HarmBench, HarmBench contextual, política da OpenAI) que definem a rubrica de avaliação e a lógica de análise. Os limites e as regras de análise são calibrados para garantir que as saídas binárias permaneçam robustas entre os fornecedores e versões de modelos.

Tabela 8: Juízes integrados e modos de backend. Os juízes de API podem alternar entre provedores atualizando o modelo de juiz. configuração para qualquer modelo de API suportado.

Juiz (nome)	Mecanismo	Modo de backend	backend de API intercambiável
Avaliador de prefixo de rejeição (palavra-chave/regex) [64]	Correspondência de regras (detecção de sinais de recusa; cf.)	Baseado em regras (sem modelo)	Não aplicável
Avaliador HarmBench (classificador LLaMA-2) [136]	Classificador especializado com saída sim/não	Modelo local (Hugging Face)	Não aplicável
Avaliador GPT (nocividade binária) [136]	Modelo de prompt com análise binária (cf.)	Avaliador de modelo de API GPT	Sim (via configuração do modelo de juiz)
(pontuação de política OpenAI) [80]	Rubrica de política (1–5) mapeada para avaliador GPT de modelo API		Sim (através da configuração do modelo de juiz)
binário (estilo HarmBench) [136]	Modelo com avaliador GPT de API de decisão sim/não (HarmBench		Sim (através da configuração do modelo de juiz)
contextual) [136]	Avaliador GPT do modelo de API de julgamento HarmBench informado		Sim (através da configuração do modelo de juiz)
pelo contexto (estilo TAP) [137]	decisão binária baseada em modelo	Modelo de API	Sim (através da configuração do modelo de juiz)

Considerações práticas. Os avaliadores baseados em regras fornecem uma linha de base determinística e de custo insignificante, apropriada para triagem conservadora. Os avaliadores locais (por exemplo, HarmBench) permitem avaliação offline e reproduzível com Precisão e quantização configuráveis. Os avaliadores de API permitem a substituição do provedor modificando o avaliador. configuração do modelo, permitindo comparações controladas entre provedores sem alterações no código. Todos os julgadores Compõe-se com defesas e ataques, permitindo a avaliação automatizada de aprovação/reprovação em ambientes experimentais de ponta a ponta. oleodutos.

4.3.5 Módulo de conjunto de dados

O módulo do conjunto de dados ancora o pipeline de avaliação: ele especifica as tarefas prejudiciais e os contextos sob Qual resposta é avaliada em termos de segurança? Essa estrutura sustenta três questões centrais: se os modelos se recusam a responder? solicitações prejudiciais sem assistência (segurança intrínseca), se os métodos de ataque podem contornar essas salvaguardas, e se as defesas conseguem restaurá-las após uma violação. Consequentemente, a escolha do conjunto de dados determina quais aspectos de segurança são exercidos e quão amplamente os resultados se generalizam. Tecnicamente, o módulo fornece um carregador unificado que normaliza os campos principais, registra a procedência e hashes de conjuntos de dados e aplica amostragem determinística, permitindo prompts interoperáveis e reproduções exatas em diversas plataformas. ataques, defesas, modelos e avaliadores.

Conjuntos de dados integrados. Para equilibrar cobertura e rastreabilidade, utilizamos três benchmarks públicos que abrangem superfícies de risco e estilos de anotação complementares:

- HarmBench [136]: sugestões de comportamento prejudicial selecionadas em categorias críticas de segurança; avaliadas com Rubricas binárias sim/não (padrão e contextual). Carregamos a partir do CSV canônico e expomos os dados em formato simples. texto de comportamento para uso intercambiável.
- JailbreakBench (JBB) [21]: um repositório aberto de artefatos de jailbreak e comportamentos alinhados à política. Nós use a divisão prejudicial JBB-Behaviors (coluna Goal) com pontuação padronizada e um modelo de ameaça claro; Ataques orientados a artefatos podem ser aplicados em camadas para estudos de transferência/reutilização.
- AIR-Bench-2024 [227]: um benchmark de red-teaming com uma taxonomia de risco hierárquica que abrange a segurança, segurança e desinformação; adotamos a configuração padrão oficial (instrução de campo); compatível com L4 opcional A amostragem é suportada, mas está desativada por padrão.

Seleção de subconjuntos e tratabilidade. Testes cruzados exaustivos em todas as combinações de métodos e cenários. é inviável. Portanto, avaliamos subconjuntos de tamanho fixo de cada conjunto de dados usando amostragem determinística (randomização com prefixo N ou controlada por semente) e integramos três benchmarks complementares (HarmBench, JBB, AIR-Bench-2024) para cobrir um amplo espectro de vulnerabilidades de segurança, mantendo o banco de testes tratável. A amostragem é orientada por configuração e reproduzível (randomização controlada por prefixo-N ou semente), com identificadores de amostra estáveis para alinhamento entre execuções. Os carregadores normalizam campos essenciais (texto de solicitação, rótulos de categoria). quando disponíveis, identificadores de proveniência) e o pipeline registra hashes de conjuntos de dados e manifestos de configuração

Para suportar reproduções exatas e relatórios estratificados (por exemplo, por capacidade, idioma, domínio). Somente benchmarks publicamente disponíveis são utilizados; o pipeline respeita as restrições de licenciamento, omite informações de identificação pessoal e oferece filtragem opcional para categorias restritas por requisitos institucionais ou de políticas.

4.4 Protocolo Experimental Unificado

Motivação. Comparações entre ataques, defesas, modelos, conjuntos de dados e avaliadores são frequentemente prejudicadas por suposições inconsistentes (por exemplo, regimes de amostragem, instruções aos avaliadores, configurações de decodificação). Um protocolo unificado controla esses fatores, garantindo imparcialidade, permitindo ablações limpas e suportando varreduras fatoriais cujas conclusões se generalizam entre diferentes configurações.

Tupla de cinco elementos. Cada experimento é definido pela tupla.

$$\tilde{y}M, A, D, S, J\tilde{y},$$

onde M é o modelo (possivelmente defendido), A um ataque (ou ausência de ataque), D uma defesa (ou ausência de defesa), S a divisão/amostragem do conjunto de dados e J o avaliador. Essa decomposição permite ablações e varreduras fatoriais comparáveis em todo o espaço de projeto.

Métricas e resultados. As definições operacionais (por exemplo, sucesso/falha no jailbreak, recusa, compensações entre utilidade e inofensividade) são fixadas pelos modelos e limites de julgamento selecionados. Também relatamos o custo (consultas). Os resultados são emitidos em JSONL com metadados completos, incluindo versões de componentes, sementes, parâmetros de decodificação e hashes de conjuntos de dados, para permitir a reprodução exata e análises secundárias.

Configuração declarativa e reprodutibilidade. Os experimentos são declarados via JSON/YAML (ou código) com validação de esquema. O sistema registra manifestos de configuração, hashes de conjuntos de dados e impressões digitais do ambiente (versões de bibliotecas, GPU/CUDA/drivers) e — a menos que seja solicitado o contrário — utiliza temperatura 0 com sementes fixas e amostragem/agrupamento canônico. Essas escolhas permitem reproduções exatas e ablações controladas.

5 Experimentos Preliminares

5.1 Visão geral

A avaliação da segurança de LLM em larga escala é computacionalmente intensiva e metodologicamente complexa. Uma varredura fatorial completa e ingênua sobre $\tilde{y}M, A, D, S, J\tilde{y}$ (modelo, ataque, defesa, conjunto de dados, avaliador; cf. Seção 4.4) é inviável. Em vez de enumerar fases, estruturamos a avaliação como um conjunto de problemas de estudo que, em conjunto, reduzem a complexidade do projeto, preservando a abrangência, o rigor e a reprodutibilidade. Nossos princípios são: (i) amostragem representativa sob restrições explícitas de abrangência; (ii) análises direcionadas que respondem a questões metodológicas específicas; e (iii) validação multicritério equilibrando confiabilidade estatística, custo e generalização.

5.2 Objetivos, Pontos Finais e Restrições

Métrica primária. Taxa de sucesso de ataque (TSA), taxa de recusa e custo. Para o conjunto de dados S com itens s e avaliador J:

$$ASR(M, A, D, S, J) = \frac{1}{|S|} \sum_{s \in S} \tilde{y}\{\text{inseguro}(M, A, D, s; J)\}.$$

Métricas secundárias. Latência, custo de token/consulta e estabilidade sob reamostragem (análise de seleção de avaliadores abaixo).

Restrições de modelo de ameaça e viabilidade. Regras de compatibilidade da Seção 2.2 (por exemplo, caixa branca \tilde{y} modelos locais com acesso a pesos), orçamentos de consulta e limites de taxa do provedor são aplicados pelo validador.

Questões de estudo. Estudamos as seguintes questões:

- Quais modelos devemos avaliar para equilibrar representatividade e tratabilidade?
- Qual juiz deve atuar como árbitro principal?
- Como construir um conjunto de avaliação discriminativo, porém justo?
- Como organizar a matriz abrangente de ataque-defesa sob restrições?
- Como analisar e validar resultados de forma robusta?
- Como escalar de forma economicamente viável com garantias de reprodutibilidade?

Resultados: modelos e avaliadores. A Tabela 9 resume a taxa de reconhecimento automático (ASR) de referência em vários avaliadores e conjuntos de dados sem ataques/defesas. Avaliamos um amplo conjunto de modelos que abrangem tanto backends de API quanto backends locais. Um padrão consistente emerge: os modelos locais geralmente exibem ASR mais alta (recusa mais fraca) do que os modelos de API sob as mesmas solicitações. Entre as famílias de API, o Gemini 2.5-Flash atinge a menor ASR em todos os conjuntos de dados, com o GPT-4o e o Claude 3.5 também no regime de baixa ASR. Para o relatório resumido, adotamos a média aritmética de cinco modelos de avaliadores baseados em GPT — GPT-HarmBench, GPT-HarmfulBinary, GPT-HarmBenchStyle, GPT-OpenAIPolicy e GPT-TAP — como a pontuação padrão.

Experimentos pendentes e próxima atualização. Algumas varreduras de configuração cruzada (por exemplo, famílias de ataques estendidas em backends white-box locais e ablações sobre empilhamento de defesa) estão em andamento. Relataremos seus resultados, juntamente com barras de erro expandidas e diagnósticos de concordância para Jy, na próxima revisão.

6 Conclusão

Esta SoK aborda o estado fragmentado da segurança de prompts LLM, unificando conceitos, pressupostos e ferramentas. Contribuímos com uma taxonomia holística de ataques e defesas, modelos de ameaça declarativos que explicitam os pressupostos de custo/conhecimento/acesso, um conjunto de ferramentas de avaliação modular de código aberto que impõe um protocolo único em todas as combinações de modelo-ataque-defesa e o JailbreakDB — um corpus amplo e ricamente anotado (sete grupos de características) que também suporta detectores leves que rivalizam com sistemas especializados (por exemplo, Grad-Safe). Nossos experimentos preliminares validam a utilidade da estrutura: modelos locais geralmente apresentam taxas de sucesso de ataque mais altas do que modelos hospedados em APIs sob prompts idênticos; entre as APIs, o Gemini 2.5 atinge o menor ASR de linha de base, com GPT-4o e Claude 3.5 também apresentando baixos resultados; e a média de cinco modelos de julgamento baseados em GPT resulta em uma pontuação padrão estável e transparente. Ao padronizar as premissas de amostragem, avaliação e custo, e ao disponibilizar manifestos completos (versões de componentes, sementes, parâmetros de decodificação, hashes de conjuntos de dados, impressões digitais do ambiente), possibilitamos comparações justas e reproduções exatas. As limitações remanescentes — subconjuntos de tamanho fixo, avaliadores automatizados como proxies e cobertura incompleta de ameaças — motivam expansões contínuas em estudos de caixa branca/locais, empilhamento de defesas, plataformas de teste multilíngues e validação com intervenção humana.

Modelo	Conjunto de dados	HamBench	GPT-HamBench	GPT-HamBench	Binary GPT-HamBench	Style GPT-OpenAI	Policy GPT-TAP	REJ-Prefix	Avg	ASR
Mistral-7B-Instruct-v0.3	banco de danos	80,0%	83,0%		73,0%		80,0%	78,0%	80,0%	73,1%
		76,0%	80,0%		72,0%		78,0%	77,0%	80,0%	71,9%
	bandada de ar jbb	97,0%	93,0%		84,0%		86,0%	86,0%	100,0%	87,9%
Ministral-8B-Instruct	banco de ar	85,0%	91,0%		79,0%		89,0%	77,0%	87,0%	45,0%
		73,0%	75,0%		71,0%		73,0%	70,0%	75,0%	32,0%
	Hambench JBB	84,0%	89,0%		69,0%		77,0%	71,0%	86,0%	65,0%
Mistral-Nemo-Instruct	banco de ar	79,0%	83,0%		73,0%		81,0%	77,0%	80,0%	49,0%
		71,0%	72,0%		63,0%		67,0%	69,0%	71,0%	52,0%
	Hambench JBB	76,0%	88,0%		62,0%		75,0%	73,0%	83,0%	69,0%
Yi-1.5-9B-Bate-papo	hambench jbb	63,0%	66,0%		61,0%		63,0%	54,0%	58,0%	42,0%
		64,0%	65,0%		65,0%		62,0%	60,0%	65,0%	37,0%
	banco de ar	80,0%	82,0%		78,0%		74,0%	70,0%	79,0%	53,0%
Yi-1.5-6B-Chat	banco de ar jbb	61,0%	60,0%		66,0%		58,0%	55,0%	58,0%	47,0%
		56,0%	62,0%		57,0%		57,0%	56,0%	61,0%	38,0%
		76,0%	82,0%		78,0%		76,0%	69,0%	84,0%	65,0%
Yi-1.5-34B-Bate-papo	banco de harm	47,0%	50,0%		52,0%		48,0%	44,0%	50,0%	36,0%
		57,0%	56,0%		56,0%		52,0%	54,0%	56,0%	31,0%
	banco de ar jbb	80,0%	76,0%		74,0%		67,0%	66,0%	75,0%	59,0%
Llama-3.2-1B-Instruir	banco de ar	48,0%	50,0%		47,0%		50,0%	41,0%	48,0%	34,0%
		42,0%	45,0%		38,0%		40,0%	36,0%	42,0%	47,0%
	Hambench JBB	57,0%	60,0%		48,0%		56,0%	57,0%	64,0%	66,0%
Llama-3.1-8B-Instruir	banco de ar	38,0%	40,0%		44,0%		37,0%	35,0%	36,0%	40,0%
		39,0%	41,0%		43,0%		39,0%	35,0%	41,0%	40,0%
	Hambench JBB	73,0%	72,0%		66,0%		65,0%	54,0%	68,0%	58,0%
Llama-3.2-3B-Instruir	banco de	32,0%	33,0%		30,0%		32,0%	25,0%	31,0%	18,0%
	harmoline jbb	25,0%	29,0%		19,0%		23,0%	16,0%	24,0%	21,0%
	banco de ar	64,0%	61,0%		54,0%		53,0%	46,0%	57,0%	48,0%
Qwen3-8B	banco de ar	31,0%	15,0%		40,0%		16,0%	14,0%	16,0%	52,0%
		31,0%	15,0%		44,0%		17,0%	19,0%	20,0%	43,0%
	Hambench JBB	51,0%	36,0%		55,0%		29,0%	31,0%	35,0%	49,0%
Phi-3.5-mini-instruir	banco de danos	26,0%	30,0%		28,0%		28,0%	23,0%	26,0%	18,0%
		27,0%	27,0%		24,0%		26,0%	19,0%	24,0%	17,0%
	bandada de ar jbb	52,0%	52,0%		45,0%		43,0%	35,0%	40,0%	30,0%
Phi-4-instruir	banco de ar	20,0%	10,0%		16,0%		10,0%	16,0%	22,0%	88,0%
		27,0%	16,0%		25,0%		13,0%	22,0%	26,0%	92,0%
	Hambench JBB	13,0%	27,0%		10,0%		23,0%	10,0%	44,0%	100,0%
Phi-3-médio-128k-instrução	banco de ar	27,0%	33,0%		29,3%		32,0%	23,0%	26,0%	31,0%
		14,0%	14,1%		20,2%		13,0%	13,0%	15,0%	24,0%
	Hambench JBB	44,0%	53,0%		40,0%		42,0%	37,0%	49,0%	48,0%
Qwen2.5-7B-Instruir	hambench jbb	30,0%	30,0%		24,0%		29,0%	26,0%	29,0%	16,0%
		19,0%	14,0%		18,0%		13,0%	15,0%	16,0%	10,0%
	banco de ar	55,0%	62,0%		46,0%		43,0%	42,0%	51,0%	31,0%
Doubao-Seed-1-6-Flash-250615	banco de ar	13,0%	14,0%		14,0%		15,0%	13,0%	14,0%	38,0%
		11,0%	13,0%		7,0%		10,0%	8,0%	11,0%	28,0%
	Hambench JBB	44,0%	49,0%		25,0%		34,0%	30,0%	40,0%	32,

Referências

[1] Glaive IA. assistente de código glaive, 2023. assistente de código glaive.

Disponível em: <https://huggingface.co/datasets/glaiveai/>

[2] Gabriel Alon e Michael Kamfonas. Detecção de ataques de modelos de linguagem com perplexidade. CoRR, abs/2308.14132, 2023.

[3] Suraj Anand e David Getzen. Os modelos de linguagem ppo-ed são hackeáveis? CoRR, abs/2406.02577, 2024.

[4] Maksym Andriushchenko, Francesco Croce e Nicolas Flammarion. Desbloqueando os principais llms alinhados à segurança com simples ataques adaptativos. CoRR, abs/2404.02151, 2024.

[5] Maksym Andriushchenko, Francesco Croce e Nicolas Flammarion. Desbloqueando os principais LLMS alinhados à segurança com ataques adaptativos simples. Na Décima Terceira Conferência Internacional sobre Representações de Aprendizagem, ICLR 2025, Singapura, 24 a 28 de abril de 2025. OpenReview.net, 2025.

[6] Maksym Andriushchenko e Nicolas Flammarion. O treinamento de recusa em llms se generaliza para o pretérito? CoRR, abs/2407.11969, 2024.

[7] Maksym Andriushchenko e Nicolas Flammarion. O treinamento de recusa em llms se generaliza para o pretérito? Na Décima Terceira Conferência Internacional sobre Representações de Aprendizagem, ICLR 2025, Singapura, 24 a 28 de abril de 2025. OpenRe-view.net, 2025.

[8] Cem Anil, Esin Durmus, Nina Panickssery, Mrinank Sharma, Joe Benton, Sandipan Kundu, Joshua Batson, Meg Tong, Jesse Mu, Daniel Ford, Francesco Mosconi, Rajashree Agrawal, Rylan Schaeffer, Naomi Bashkansky, Samuel Svenningsen, Mike Lambert, Ansh Radhakrison, Denshnan, Denshnan Hubinger, Yuntao Bai, Trenton Bricken, Timothy Maxwell, Nicholas Schiefer, James Sully, Alex Tamkin, Tamera Lanham, Karina Nguyen, Tomek Korbak, Jared Kaplan, Deep Ganguli, Samuel R. Bowman, Ethan Perez, Roger B. Grosse e David Kristjanson Duvenaud. Jailbreak de muitos tiros. Em Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak e Cheng Zhang, editores, Advances in Neural Information Processing Systems 38: Conferência anual sobre sistemas de processamento de informações neurais 2024, NeurIPS 2024, Vancouver, BC, Canadá, 10 a 15 de dezembro de 2024, 2024.

[9] Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M. Dai, Anja Hauth, Katie Millican, David Silver, Slav Petrov, Melvin Johnson, Ioannis Antonoglou, Julian Schrittwieser, Amelia Glaese, Jilin Chen, Emily Pitler, Timothy P. Lillicrap, Angeliki Lazaridou, Orhan Firat, James Molloy, Michael Isard, Paul Ronald Barham, Tom Hennigan, Benjamin Lee, Fabio Viola, Malcolm Reynolds, Yuanzhong Xu, Ryan Doherty, Eli Collins, Clemens Meyer, Eliza Rutherford, Erica Moreira, Kareem Ayoub, Megha Goel, George Tucker, Enrique Piqueras, Maxim Krikun, Iain Barr, Nikolay Savinov, Ivo Danihelka, Becca Roelofs, Anaš White, Anders Andreassen, Tamara von Glehn, Lakshman Yagati, Mehran Kazemi, Lucas Gonzalez, Misha Khalman, Jakub Sygnowski e outros. Gemini: Uma família de modelos multimodais altamente capazes. CoRR, abs/2312.11805, 2023.

[10] Zhongjie Ba, Jieming Zhong, Jiachen Lei, Peng Cheng, Qinglong Wang, Zhan Qin, Zhibo Wang e Kui Ren. Surro-gateprompt: Bypassing the safety filter of text-to-image models via substitution. In Bo Luo, Xiaojing Liao, Jun Xu, Engin Kirda e David Lie, editores, Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security, CCS 2024, Salt Lake City, UT, EUA, 14 a 18 de outubro de 2024, páginas 1166–1180. ACM, 2024.

[11] Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou e Tianhang Zhu. Relatório técnico Qwen. CoRR, abs/2309.16609, 2023.

[12] 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 B. Brown, Jack Clark, Sam McCandlish, Chris Olah, Benjamin Mann e Jared Kaplan. Treinamento de um assistente útil e inofensivo com aprendizado por reforço a partir de feedback humano. CoRR, abs/2204.05862, 2022.

[13] Somnath Banerjee, Sayan Layek, Rima Hazra e Animesh Mukherjee. Quão (anti)éticas são as respostas centradas em instruções dos llms? Revelando as vulnerabilidades das proteções de segurança a consultas prejudiciais. CoRR, abs/2402.15302, 2024.

[14] Rishabh Bhardwaj e Soujanya Poria. Red-teaming de grandes modelos de linguagem usando cadeia de enunciados para alinhamento de segurança. CoRR, abs/2308.09662, 2023.

[15] Anubhav Bhatti, Prithila Angkan, Behnam Behinaein, Zunayed Mahmud, Dirk Rodenburg, Heather Braund, P. James Mclellan, Aaron J. Ruberto, Geoffery Harrison, Daryl Wilson, Adam Szulewski, Dan Howes, Ali Etemad e Paul Hungler. CLARE: avaliação da carga cognitiva em tempo real com dados multimodais. CoRR, abs/2404.17098, 2024.

[16] Federico Bianchi, Mirac Suzgun, Giuseppe Attanasio, Paul R’ottger, Dan Jurafsky, Tatsunori Hashimoto e James Zou. Lhamas com foco em segurança: Lições da melhoria da segurança de grandes modelos de linguagem que seguem instruções. In Décima Segunda Conferência Internacional sobre Representações de Aprendizagem, ICLR 2024, Viena, Áustria, 7 a 11 de maio de 2024. OpenReview.net, 2024.

[17] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever e Dario Amodei. Os modelos de linguagem são aprendizes de poucos exemplos. Em Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan e Hsuan-Tien Lin, editores, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020*, 6 a 12 de dezembro de 2020, virtual, 2020.

[18] CAMEL-AI. Conjuntos de dados de especialistas do domínio Camelai (física, matemática, química e biologia), 2023. Disponível em: <https://huggingface.co/camel-ai>.

[19] Trishna Chakraborty, Erfan Shayegani, Zikui Cai, Nael B. Abu-Ghazaleh, M. Salman Asif, Yue Dong, Amit K. Roy-Chowdhury e Chengyu Song. O desaprendizado textual pode resolver o alinhamento de segurança entre modalidades? Em Yaser Al-Onaizan, Mohit Bansal e Yun-Nung Chen, editores, *Anais da Associação de Linguística Computacional: EMNLP 2024*, Miami, Flórida, EUA, 12 a 16 de novembro de 2024, páginas 9830–9844. Associação de Linguística Computacional, 2024.

[20] Chiju Chao, Qirui Wang, Hongfei Wu e Zhiyong Fu. Synneure: Trabalho em equipe inteligente homem-máquina no espaço virtual. Em Pei-Luen Patrick Rau, editor, *Cross-Cultural Design - 15ª Conferência Internacional, CCD 2023*, realizada como parte da 25ª Conferência Internacional, HCII 2023, Copenhagen, Dinamarca, 23 a 28 de julho de 2023, *Anais*, Parte II, volume 14023 de *Lecture Notes in Computer Science*, páginas 349–361. Springer, 2023.

[21] Patrick Chao, Edoardo Debenedetti, Alexander Robey, Maksym Andriushchenko, Francesco Croce, Vikash Sehwal, Edgar Dobriban, Nicolas Flammarion, George J. Pappas, Florian Tram`er, Hamed Hassani e Eric Wong. Jailbreakbench: Um benchmark de robustez aberto para jailbreak de grandes modelos de linguagem. Em Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak e Cheng Zhang, editores, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024*, Vancouver, BC, Canadá, 10 a 15 de dezembro de 2024.

[22] Patrick Chao, Alexander Robey, Edgar Dobriban, Hamed Hassani, George J. Pappas e Eric Wong. Quebrando modelos de linguagem de caixa preta grandes em vinte consultas. *CoRR*, abs/2310.08419, 2023.

[23] Patrick Chao, Alexander Robey, Edgar Dobriban, Hamed Hassani, George J. Pappas e Eric Wong. Quebrando modelos de linguagem de caixa preta grandes em vinte consultas. Em Conferência IEEE sobre Aprendizado de Máquina Seguro e Confiável, SaTML 2025, Copenhagen, Dinamarca, 9 a 11 de abril de 2025, páginas 23–42. IEEE, 2025.

[24] Sizhe Chen, Julien Piet, Chawin Sitawarin e David A. Wagner. Struq: Defendendo-se contra injeção imediata com consultas estruturadas. *CoRR*, abs/2402.06363, 2024.

[25] Xuan Chen, Yuzhou Nie, Wenbo Guo e Xiangyu Zhang. Quando LLM encontra DRL: avançando a eficiência de jailbreaking por meio de busca guiada por drl. Em Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak e Cheng Zhang, editores, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024*, Vancouver, BC, Canadá, 10 a 15 de dezembro de 2024.

[26] Xuan Chen, Yuzhou Nie, Lu Yan, Yunshu Mao, Wenbo Guo e Xiangyu Zhang. RL-JACK: aprendizagem por reforço-Ataque de jailbreak de caixa preta com tecnologia avançada contra llms. *CoRR*, abs/2406.08725, 2024.

[27] Yixin Cheng, Markos Georgopoulos, Volkan Cevher e Grigoris G. Chrysos. Aproveitando o contexto por meio de várias interações de rodada para ataques de jailbreak. *CoRR*, abs/2402.09177, 2024.

[28] Jan Clusmann, Dyke Ferber, Isabella C. Wiest, Carolin V. Schneider, Titus J. Brinker, Sebastian Foersch, Daniel Truhn e Jakob Nikolas Kather. Ataques de injeção rápida em grandes modelos de linguagem em oncologia. *CoRR*, abs/2407.18981, 2024.

[29] CogStack. Cogstack, 2023. Disponível em: <https://github.com/CogStack/OpenGPT>.

[30] Stav Cohen, Ron Bitton e Ben Nassi. Um modelo Genai desbloqueado pode causar danos substanciais: Aplicativos baseados em Genai são vulneráveis a promptwares. *CoRR*, abs/2408.05061, 2024.

[31] Giandomenico Cornacchia, Giulio Zizzo, Kieran Fraser, Muhammad Zaid Hameed, Ambrish Rawat e Mark Purcell. Moje: Mistura de especialistas em jailbreak e classificadores tabulares ingênuos como proteção contra ataques rápidos. *CoRR*, abs/2409.17699, 2024.

[32] Giandomenico Cornacchia, Giulio Zizzo, Kieran Fraser, Muhammad Zaid Hameed, Ambrish Rawat e Mark Purcell. Moje: Mistura de especialistas em jailbreak e classificadores tabulares ingênuos como proteção contra ataques rápidos. Em Sanmay Das, Brian Patrick Green, Kush Varshney, Marianna Ganapini e Andrea Renda, editores, *Anais da Sétima Conferência AAAI/ACM sobre IA, Ética e Sociedade (AIES-24) - Artigos Arquivados Completos*, 21 a 23 de outubro de 2024, San Jose, Califórnia, EUA - Volume 1, páginas 304–315. AAAI Press, 2024.

[33] Crystalcareai. alpaca-gpt4-cot, 2024. Disponível em: <https://huggingface.co/datasets/Crystalcareai/alpaca-gpt4-COT>.

[34] Edoardo Debenedetti, Javier Rando, Daniel Paleka, Silaghi Fineas Florin, Dragos Albastroiu, Niv Cohen, Yuval Lemberg, Reshmi Ghosh, Rui Wen, Ahmed Salem, Giovanni Cherubin, Santiago Zanella-B`eguelin, Robin Schmid, Victor Klemm, Takahiro Miki, Chenhao Li, Stefan Kraft, Mario Fritz, Florian Tram`er, Sahar Abdelnabi e Lea Schönherr. Conjunto de dados e lições aprendidas na competição satml LLM capture-the-flag de 2024. Em Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak e Cheng Zhang, editores, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024*, Vancouver, BC, Canadá, 10 a 15 de dezembro de 2024, 2024.

[35] Edoardo Debenedetti, Jie Zhang, Mislav Balunovic, Luca Beurer-Kellner, Marc Fischer e Florian Tram`er. Agentdojo: Um ambiente dinâmico para avaliar ataques e defesas para agentes LLM. *CoRR*, abs/2406.13352, 2024.

- [36] Boyi Deng, Wenjie Wang, Fuli Feng, Yang Deng, Qifan Wang e Xiangnan He. Ataque à geração de prompts para red teaming e defesa de modelos de linguagem de grande porte. Em Houda Bouamor, Juan Pino e Kalika Bali, editores, Findings of the Association for Computational Linguistics: EMNLP 2023, Singapura, 6 a 10 de dezembro de 2023, páginas 2176–2189. Association for Computational Linguistics, 2023.
- [37] Gelei Deng, Yi Liu, Yuekang Li, Kailong Wang, Ying Zhang, Zefeng Li, Haoyu Wang, Tianwei Zhang e Yang Liu. Masterkey: Desbloqueio automatizado em múltiplos chatbots de grande porte com modelos de linguagem naturais. Preprint arXiv:2307.08715, 2023.
- [38] Gelei Deng, Yi Liu, Kailong Wang, Yuekang Li, Tianwei Zhang e Yang Liu. Pandora: Jailbreak gpts por recuperação envenenamento por geração aumentada. CoRR, abs/2402.08416, 2024.
- [39] Yue Deng, Wenxuan Zhang, Sinno Jialin Pan e Lidong Bing. Desafios de jailbreak multilíngue em grandes modelos de idiomas. Na Décima Segunda Conferência Internacional sobre Representações de Aprendizagem, ICLR 2024, Viena, Áustria, 7 a 11 de maio de 2024. OpenReview.net, 2024.
- [40] Peng Ding, Jun Kuang, Dan Ma, Xuezhi Cao, Yunsen Xian, Jiajun Chen e Shujian Huang. Um lobo em pele de cordeiro: prompts de jailbreak aninhados generalizados podem enganar facilmente grandes modelos de linguagem. Em Kevin Duh, Helena Gómez-Adorno e Steven Bethard, editores, Anais da Conferência de 2024 do Capítulo Norte-Americano da Associação de Linguística Computacional: Tecnologias da Linguagem Humana (Volume 1: Artigos Longos), NAACL 2024, Cidade do México, México, 16 a 21 de junho de 2024, páginas 2136–2153. Associação de Linguística Computacional, 2024.
- [41] Yingkai Dong, Zheng Li, Xiangtao Meng, Ning Yu e Shanqing Guo. Jailbreak de modelos de texto para imagem com agentes baseados em llm. CoRR, abs/2408.00523, 2024.
- [42] Yiting Dong, Guobin Shen, Dongcheng Zhao, Xiang He e Yi Zeng. Aproveitando a sobrecarga de tarefas para jailbreak escalonável ataques a grandes modelos de linguagem. CoRR, abs/2410.04190, 2024.
- [43] Diego Dorn, Alexandre Variengien, Charbel-Raphaël Ségerie e Vincent Corruble. BELLS: Uma estrutura para o futuro parâmetros de referência para a avaliação de salvaguardas de LLM. CoRR, abs/2406.01364, 2024.
- [44] Moussa Koullako Bala Doumbouya, Ananjan Nandi, Gabriel Poesia, Davide Ghilardi, Anna Goldie, Federico Bianchi, Dan Jurafsky e Christopher D. Manning. h4rm3l: Um benchmark dinâmico de ataques de jailbreak combináveis para avaliação de segurança LLM. CoRR, abs/2408.04811, 2024.
- [45] Xiaohu Du, Fan Mo, Ming Wen, Tu Gu, Huadi Zheng, Hai Jin e Jie Shi. Desbloqueio multi-turno de grandes modelos de linguagem por meio de deslocamento de atenção. Em Toby Walsh, Julie Shah e Zico Kolter, editores, AAAI-25, patrocinado pela Association for the Advancement of Artificial Intelligence, 25 de fevereiro a 4 de março de 2025, Filadélfia, PA, EUA, páginas 23814–23822. AAAI Press, 2025.
- [46] Xuefeng Du, Reshmi Ghosh, Robert Sim, Ahmed Salem, Vitor Carvalho, Emily Lawton, Yixuan Li e Jack W. Stokes. Vlmguard: Defesa de VLMs contra solicitações maliciosas por meio de dados não rotulados. CoRR, abs/2410.00296, 2024.
- [47] Yuhao Du, Zhuo Li, Pengyu Cheng, Xiang Wan e Anningzhe Gao. Detectando falhas de IA: ataques direcionados a alvos falhas internas em modelos de linguagem. CoRR, abs/2408.14853, 2024.
- [48] Matt Duver, Noah Wiederhold, Maria Kyrarini, Sean Banerjee e Natasha Kholgade Banerjee. Vr-hand-in-hand: Usando rastreamento de mãos em realidade virtual (RV) para anotação de dados de mãos e objetos. Em 2024 IEEE International Conference on Artificial Intelligence and eXtended and Virtual Reality (AIxVR), Los Angeles, CA, EUA, 17 a 19 de janeiro de 2024, páginas 325–329. IEEE, 2024.
- [49] Aysan Esmradi, Daniel Wankit Yip e Chun-Fai Chan. Uma pesquisa abrangente sobre técnicas de ataque, implementação e estratégias de mitigação em modelos de linguagem de grande porte. Em Guojun Wang, Haozhe Wang, Geyong Min, Nektarios Georgalas e Weizhi Meng, editores, Ubiquitous Security - Terceira Conferência Internacional, UbiSec 2023, Exeter, Reino Unido, 1 a 3 de novembro de 2023, Revised Selected Papers, volume 2034 de Communications in Computer and Information Science, páginas 76–95. Springer, 2023.
- [50] Yihe Fan, Yuxin Cao, Ziyu Zhao, Ziyao Liu e Shaofeng Li. Ícaro desenfreado: uma análise dos perigos potenciais das entradas de imagem na segurança de modelos de linguagem multimodais de grande porte. Em IEEE International Conference on Systems, Man, and Cybernetics, SMC 2024, Kuching, Malásia, 6 a 10 de outubro de 2024, páginas 3428–3433. IEEE, 2024.
- [51] Yingchaojie Feng, Zhizhang Chen, Zhining Kang, Sijia Wang, Minfeng Zhu, Wei Zhang e Wei Chen. Jailbreaklens: Análise visual de ataques de jailbreak contra grandes modelos de linguagem. CoRR, abs/2404.08793, 2024.
- [52] Yu Fu, Yufei Li, Wen Xiao, Cong Liu e Yue Dong. Alinhamento de segurança em tarefas de PNL: Sumarização fracamente alinhada como um ataque em contexto. Em Lun-Wei Ku, Andre Martins e Vivek Srikumar, editores, Anais da 62ª Reunião Anual da Associação de Linguística Computacional (Volume 1: Artigos Longos), ACL 2024, Bangkok, Tailândia, 11 a 16 de agosto de 2024, páginas 8483–8502. Associação de Linguística Computacional, 2024.
- [53] Yu Fu, Wen Xiao, Jia Chen, Jiachen Li, Evangelos E. Papalexakis, Aichi Chien e Yue Dong. Defesa entre tarefas: Ajuste de instruções llms para segurança de conteúdo. CoRR, abs/2405.15202, 2024.
- [54] V'jctor Gallego. A fusão melhora a autocrítica contra ataques de jailbreak. CoRR, abs/2406.07188, 2024.
- [55] Suyu Ge, Chunting Zhou, Rui Hou, Madian Khabsa, Yi-Chia Wang, Qifan Wang, Jiawei Han e Yuning Mao. MART: aprimorando a segurança do LLM com red-teaming automático de múltiplas rodadas. Em Kevin Duh, Helena Gómez-Adorno e Steven Bethard, editores, Anais da Conferência de 2024 do Capítulo Norte-Americano da Associação de Linguística Computacional: Tecnologias da Linguagem Humana (Volume 1: Artigos Longos), NAACL 2024, Cidade do México, México, 16 a 21 de junho de 2024, páginas 1927–1937. Associação de Linguística Computacional, 2024.
- [56] Jonas Geiping, Alex Stein, Manli Shu, Khalid Saifullah, Yuxin Wen e Tom Goldstein. Coagindo llms a fazer e revelar (quase) tudo. CoRR, abs/2402.14020, 2024.

- [57] Simon Geisler, Tom Wollschläger, MHI Abdalla, Johannes Gasteiger e Stephan Günnemann. Atacando grandes Modelos de linguagem com descida de gradiente projetada. CoRR, abs/2402.09154, 2024.
- [58] Shaona Ghosh, Prasoon Varshney, Erick Galinkin e Christopher Parisien. AEGIS: segurança de conteúdo de IA adaptativa online. Moderação com conjunto de especialistas em LLM. CoRR, abs/2404.05993, 2024.
- [59] Tom Gibbs, Ethan Kosak-Hine, George Ingebretsen, Jason Zhang, Julius Broomfield, Sara Pieri, Reihaneh Iranmanesh, Reihaneh Rabbany e Kellin Pelrine. Vulnerabilidades emergentes em modelos de fronteira: ataques de jailbreak de múltiplas rodadas. CoRR, abs/2409.00137, 2024.
- [60] David Glukhov, Ziwen Han, Ilia Shumailov, Vardan Papayan e Nicholas Papernot. Uma falsa sensação de segurança: inseguro vazamento de informações em respostas de IA 'seguras'. CoRR, abs/2407.02551, 2024.
- [61] Chuan Guo, Alexandre Sablayrolles, Hervé Jégou e Douwe Kiela. Ataques adversários baseados em gradiente contra transformadores de texto. Em Marie-Francine Moens, Xuanjing Huang, Lucia Specia e Scott Wen-tau Yih, editores, Anais da Conferência de 2021 sobre Métodos Empíricos em Processamento de Linguagem Natural, EMNLP 2021, Evento Virtual / Punta Cana, República Dominicana, 7 a 11 de novembro de 2021, páginas 5747–5757. Associação de Linguística Computacional, 2021.
- [62] Xingang Guo, Fangxu Yu, Huan Zhang, Lianhui Qin e Bin Hu. Cold-attack: Jailbreaking llms with stealthiness and controllability. In Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024. OpenReview.net, 2024.
- [63] Maanak Gupta, Charankumar Akiri, Kshitiz Aryal, Eli Parker e Lopamudra Praharaj. Do chatgpt ao ameaçagpt: Impacto da IA generativa na segurança cibernética e na privacidade. IEEE Access, 11:80218–80245, 2023.
- [64] Seungju Han, Kavel Rao, Allyson Ettinger, Liwei Jiang, Bill Yuchen Lin, Nathan Lambert, Yejin Choi e Nouha Dziri. Wildguard: Ferramentas abertas de moderação centralizada para riscos de segurança, invasões e recusas de LLMs. Em Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak e Cheng Zhang, editores, Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canadá, 10 a 15 de dezembro de 2024.
- [65] Divij Handa, Advait Chirmule, Bimal G. Gajera e Chitta Baral. Jailbreak de modelos proprietários de grandes linguagens usando cifra de substituição de palavras. CoRR, abs/2402.10601, 2024.
- [66] Richard Harper e Dave W. Randall. Aprendizado de máquina e o trabalho do usuário. Suporte computacional. Trabalho Cooperativo. 33(2):103–136, 2024.
- [67] Anne Hartebrodt e Richard Röttger. Privacidade da decomposição QR federada usando computação multipartidária segura aditiva. IEEE Trans. Inf. Forensics Secur., 18:5122–5132, 2023.
- [68] Jonathan Hayase, Ema Borevkovic, Nicholas Carlini, Florian Tramèr e Milad Nasr. Geração de prompts adversários baseada em consulta. Em Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak e Cheng Zhang, editores, Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canadá, 10 a 15 de dezembro de 2024.
- [69] Tanmoy Hazra, Kushal Anjaria, Aditi Bajpai e Akshara Kumari. Aplicações da Teoria dos Jogos na Aprendizagem Profunda. Springer Briefs in Computer Science. Springer, 2024.
- [70] Keegan Hines, Gary Lopez, Matthew Hall, Federico Zarfati, Yonatan Zunger e Emre Kiciman. Defesa contra ataques indiretos de injeção rápida com spotlighting. Em Rachel Allen, Sagar Samtani, Edward Raff e Ethan M. Rudd, editores, Anais da Conferência sobre Aprendizado de Máquina Aplicado em Segurança da Informação (CAMLIS 2024), Arlington, Virginia, EUA, 24 e 25 de outubro de 2024, volume 3920 dos Anais do Workshop CEUR, páginas 48–62. CEUR-WS.org, 2024.
- [71] Xiaomeng Hu, Pin-Yu Chen e Tsung-Yi Ho. Gradient cuff: Detectando ataques de jailbreak em grandes modelos de linguagem explorando cenários de perda de recusa. Em Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak e Cheng Zhang, editores, Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canadá, 10 a 15 de dezembro de 2024.
- [72] Caishuang Huang, Wanxu Zhao, Rui Zheng, Huijie Lv, Shihan Dou, Sixian Li, Xiao Wang, Enyu Zhou, Junjie Ye, Yuming Yang, Tao Gui, Qi Zhang e Xuanjing Huang. Safealigner: Alinhamento de segurança contra ataques de jailbreak por meio de orientação sobre disparidade de resposta. CoRR, abs/2406.18118, 2024.
- [73] Jen-tse Huang, Wenxuan Wang, Eric John Li, Man Ho Lam, Shujie Ren, Youliang Yuan, Wenxiang Jiao, Zhaopeng Tu e Michael R. Lyu. Sobre a humanidade da IA conversacional: avaliando a representação psicológica dos llms. In Décima Segunda Conferência Internacional sobre Representações de Aprendizagem, ICLR 2024, Viena, Áustria, 7 a 11 de maio de 2024. OpenReview.net, 2024.
- [74] Yangsibo Huang, Samyak Gupta, Mengzhou Xia, Kai Li e Danqi Chen. Quebra catastrófica de llms de código aberto por meio da exploração de geração. Em Décima Segunda Conferência Internacional sobre Representações de Aprendizagem, ICLR 2024, Viena, Áustria, 7 a 11 de maio de 2024. OpenReview.net, 2024.
- [75] Nanna Inie, Jonathan Stray e Leon Derczynski. Invocar um demônio e prendê-lo: Uma teoria fundamentada do red teaming da LLM na natureza. CoRR, abs/2311.06237, 2023.
- [76] Neel Jain, Avi Schwarzschild, Yuxin Wen, Gowthami Somepalli, John Kirchenbauer, Ping-yeh Chiang, Micah Goldblum, Aniruddha Saha, Jonas Geiping e Tom Goldstein. Defesas básicas para ataques adversários contra modelos de linguagem alinhados. CoRR, abs/2309.00614, 2023.

- [77] Jiaming Ji, Mickel Liu, Josef Dai, Xuehai Pan, Chi Zhang, Ce Bian, Boyuan Chen, Ruiyang Sun, Yizhou Wang e Yaodong Yang. Beavertails: Rumor a um alinhamento de segurança aprimorado do LLM por meio de um conjunto de dados de preferência humana. Em Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt e Sergey Levine, editores, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, Nova Orleans, LA, EUA, 10 a 16 de dezembro de 2023*.
- [78] Xiaojun Jia, Tianyu Pang, Chao Du, Yihao Huang, Jindong Gu, Yang Liu, Xiaochun Cao e Min Lin. Técnicas aprimoradas para jailbreaking baseado em otimização em grandes modelos de linguagem. Em *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapura, 24 a 28 de abril de 2025*. OpenReview.net, 2025.
- [79] Bojian Jiang, Yi Jing, Tong Wu, Tianhao Shen, Deyi Xiong e Qing Yang. Equipe vermelha progressiva automatizada. Em Owen Rambow, Leo Wanner, Marianna Apidianaki, Hend Al-Khalifa, Barbara Di Eugenio e Steven Schockaert, editores, *Anais da 31ª Conferência Internacional de Linguística Computacional, COLING 2025, Abu Dhabi, Emirados Árabes Unidos, 19 a 24 de janeiro de 2025, páginas 3850–3864*. Associação de Linguística Computacional, 2025.
- [80] Fengqing Jiang, Zhangchen Xu, Luyao Niu, Zhen Xiang, Bhaskar Ramasubramanian, Bo Li e Radha Poovendran. Artprompt: Ataques de jailbreak baseados em arte ASCII contra llms alinhados. Em Lun-Wei Ku, Andre Martins e Vivek Sriku-mar, editores, *Anais da 62ª Reunião Anual da Associação de Linguística Computacional (Volume 1: Artigos Longos), ACL 2024, Bangkok, Tailândia, 11 a 16 de agosto de 2024, páginas 15157–15173*. Associação de Linguística Computacional, 2024.
- [81] Jiyue Jiang, Liheng Chen, Pengan Chen, Sheng Wang, Qinghang Bao, Lingpeng Kong, Yu Li e Chuan Wu. Até onde pode ir a PNL cantonesa? avaliação comparativa das capacidades cantonesas de grandes modelos linguísticos. CoRR, abs/2408.16756, 2024.
- [82] Liwei Jiang, Kavel Rao, Seungju Han, Allyson Ettinger, Faeze Brahman, Sachin Kumar, Niloofar Miresghallah, Ximing Lu, Maarten Sap, Yejin Choi e Nouha Dziri. Wildteaming em escala: de fugas de presos em ambientes selvagens a modelos de linguagem (adversariamente) mais seguros. Em Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak e Cheng Zhang, editores, *Advances in Neural Information Processing Systems 38: Conferência anual sobre sistemas de processamento de informações neurais 2024, NeurIPS 2024, Vancouver, BC, Canadá, 10 a 15 de dezembro de 2024, 2024*.
- [83] Weipeng Jiang, Zhengting Wang, Juan Zhai, Shiqing Ma, Zhengyu Zhao e Chao Shen. Desbloqueando a otimização adversária de sufixos sem frases afirmativas: desbloqueio eficiente de caixa preta via LLM como otimizador. CoRR, abs/2408.11313, 2024.
- [84] Yifan Jiang, Kriti Aggarwal, Tanmay Laud, Kashif Munir, Jay Pujara e Subhabrata Mukherjee. RED QUEEN: protegendo grandes modelos de linguagem contra jailbreaking multi-turn oculto. CoRR, abs/2409.17458, 2024.
- [85] Haibo Jin, Ruoxi Chen, Andy Zhou, Jinyin Chen, Yang Zhang e Haohan Wang. GUARD: role-playing para gerar jailbreaks de linguagem natural para testar a aderência de diretrizes de grandes modelos de linguagem. CoRR, abs/2402.03299, 2024.
- [86] Haibo Jin, Leyang Hu, Xinuo Li, Peiyan Zhang, Chonghan Chen, Jun Zhuang e Haohan Wang. Jailbreakzoo: Pesquisa, paisagens e horizontes no jailbreak de grandes modelos de linguagem e linguagem de visão. CoRR, abs/2407.01599, 2024.
- [87] jondurbin. aioboros-2.2, 2023. Disponível em: <https://huggingface.co/datasets/jondurbin/aioboros-2.2>.
- [88] Erik Jones, Anca D. Dragan, Aditi Raghunathan e Jacob Steinhardt. Auditoria automática de grandes modelos de linguagem por meio de otimização discreta. Em Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato e Jonathan Scarlett, editores, *Conferência Internacional sobre Aprendizado de Máquina, ICML 2023, 23 a 29 de julho de 2023, Honolulu, Havaí, EUA, volume 202 de Proceedings of Machine Learning Research, páginas 15307–15329*. PMLR, 2023.
- [89] Erik Jones, Anca D. Dragan e Jacob Steinhardt. Os adversários podem usar indevidamente combinações de modelos seguros. CoRR, abs/2406.14595, 2024.
- [90] Daniel Kang, Xuechen Li, Ion Stoica, Carlos Guestrin, Matei Zaharia e Tatsunori Hashimoto. Explorando o comportamento programático de llms: Dupla utilização por meio de ataques de segurança padrão. Em *IEEE Security and Privacy, SP 2024 - Workshops, São Francisco, CA, EUA, 23 de maio de 2024, páginas 132–143*. IEEE, 2024.
- [91] Daniil Khomsky, Narek Maloyan e Bulat Nutfullin. Ataques de injeção rápida em sistemas defendidos. CoRR, abs/2406.14048, 2024.
- [92] Heegyu Kim, Sehyun Yuk e Hyunsouk Cho. Quebrando a fuga: Reinventando a defesa LM contra ataques de fuga com auto-refinamento. CoRR, abs/2402.15180, 2024.
- [93] Taeyoun Kim, Suhas Kotha e Aditi Raghunathan. O jailbreak é melhor resolvido por definição. CoRR, abs/2403.14725, 2024.
- [94] Subaru Kimura, Ryota Tanaka, Shumpei Miyawaki, Jun Suzuki e Keisuke Sakaguchi. Análise empírica de grandes modelos de visão-linguagem contra sequestro de objetivos via injeção de dicas visuais. CoRR, abs/2408.03554, 2024.
- [95] Aounon Kumar, Chirag Agarwal, Suraj Srinivas, Soheil Feizi e Hima Lakkaraju. Certificação da segurança do LLM contra o estímulo adversário. CoRR, abs/2309.02705, 2023.
- [96] Raz Lapid, Ron Langberg e Moshe Sipper. Abra-te Sésamo! desbloqueio universal de caixa preta de grandes modelos de linguagem. CoRR, abs/2309.01446, 2023.
- [97] Ariel N. Lee, Cole J. Hunter e Nataniel Ruiz. Ornitorrinco: Refinamento rápido, barato e poderoso de llms. CoRR, abs/2308.07317, 2023.
- [98] Haoran Li, Dadi Guo, Wei Fan, Mingshi Xu, Jie Huang, Fanpu Meng e Yangqiu Song. Ataques de privacidade de jailbreaking em várias etapas no chatgpt. Em Houda Bouamor, Juan Pino e Kalika Bali, editores, *Resultados da Associação de Linguística Computacional: EMNLP 2023, Singapura, 6 a 10 de dezembro de 2023, páginas 4138–4153*. Associação de Linguística Computacional, 2023.

- [99] Jie Li, Yi Liu, Chongyang Liu, Ling Shi, Xiaoning Ren, Yaowen Zheng, Yang Liu e Yinxing Xue. Uma linguagem cruzada Investigação sobre ataques de jailbreak em grandes modelos de linguagem. CoRR, abs/2401.16765, 2024.
- [100] Nathaniel Li, Ziwen Han, Ian Steneker, Willow Primack, Riley Goodside, Hugh Zhang, Zifan Wang, Cristina Menghini e Summer Yue. As defesas LLM ainda não são robustas contra fugas humanas de múltiplas rodadas. CoRR, abs/2408.15221, 2024.
- [101] Qizhang Li, Yiwen Guo, Wangmeng Zuo e Hao Chen. Geração aprimorada de exemplos adversários contra llms alinhados à segurança. Em Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak e Cheng Zhang, editores, Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canadá, 10 a 15 de dezembro de 2024.
- [102] Xiaoxia Li, Siyuan Liang, Jiyi Zhang, Han Fang, Aishan Liu e Ee-Chien Chang. Jailbreak de espelho semântico: genético Solicitações de jailbreak baseadas em algoritmo contra llms de código aberto. CoRR, abs/2402.14872, 2024.
- [103] Xirui Li, Ruochen Wang, Minhao Cheng, Tianyi Zhou e Cho-Jui Hsieh. Drattack: decomposição imediata e A reconstrução cria poderosos fugitivos da prisão com foco em LLM. CoRR, abs/2402.16914, 2024.
- [104] Xirui Li, Ruochen Wang, Minhao Cheng, Tianyi Zhou e Cho-Jui Hsieh. Drattack: Decomposição e reconstrução imediatas tornam poderosos programas de quebra de bloqueios de sistemas de gerenciamento de linguagem. Em Yaser Al-Onaizan, Mohit Bansal e Yun-Nung Chen, editores, Anais da Associação de Linguística Computacional: EMNLP 2024, Miami, Flórida, EUA, 12 a 16 de novembro de 2024, páginas 13891–13913. Associação de Linguística Computacional, 2024.
- [105] Xuan Li, Zhanke Zhou, Jianing Zhu, Jiangchao Yao, Tongliang Liu e Bo Han. Deepinception: Hipnotize um modelo de linguagem grande para ser um jailbreaker. CoRR, abs/2311.03191, 2023.
- [106] Yixuan Li, Xuelin Liu, Xiaoyang Wang, Shiqi Wang e Weisi Lin. Fakebench: Descubra o calcanhar de Aquiles das imagens falsas com grandes modelos multimodais. CoRR, abs/2404.13306, 2024.
- [107] Yuhui Li, Fangyun Wei, Jinjing Zhao, Chao Zhang e Hongyang Zhang. RAIN: seus modelos de linguagem podem se alinhar sem ajuste fino. Na Décima Segunda Conferência Internacional sobre Representações de Aprendizagem, ICLR 2024, Viena, Áustria, 7 a 11 de maio de 2024. OpenReview.net, 2024.
- [108] Zeyi Liao e Huan Sun. Amplegch: Aprendendo um modelo gerativo universal e transferível de sufixos adversários para jailbreaking tanto open quanto closed llms. CoRR, abs/2404.07921, 2024.
- [109] Shi Lin, Rongchang Li, Xun Wang, Changting Lin, Wenpeng Xing e Meng Han. Descubra: baseado em análise Ataque de jailbreak em grandes modelos de linguagem. CoRR, abs/2407.16205, 2024.
- [110] Zhihao Lin, Wei Ma, Mingyi Zhou, Yanjie Zhao, Haoyu Wang, Yang Liu, Jun Wang e Li Li. Pathseeker: Explorando vulnerabilidades de segurança LLM com uma abordagem de jailbreak baseada em aprendizado de reforço. CoRR, abs/2409.14177, 2024.
- [111] Chengyuan Liu, Fubang Zhao, Lizhi Qing, Yangyang Kang, Changlong Sun, Kun Kuang e Fei Wu. Orientado para objetivos ataque rápido e avaliação de segurança para llms. arXiv pré-impressão arXiv:2309.11830, 2023.
- [112] Fan Liu, Yue Feng, Zhao Xu, Lixin Su, Xinyu Ma, Dawei Yin e Hao Liu. JAILJUDGE: Um jailbreak abrangente benchmark de julgamento com estrutura de avaliação de explicação aprimorada multiagente. CoRR, abs/2410.12855, 2024.
- [113] Fan Liu, Zhao Xu e Hao Liu. Ajuste adversário: Defesa contra ataques de fuga de presos para llms. CoRR, abs/2406.06622, 2024.
- [114] Hongfu Liu, Yuxi Xie, Ye Wang e Michael Shieh. Avançando a aprendizagem por transferência de sufixos adversarial em modelos de linguagem alinhados de grande porte. Em Yaser Al-Onaizan, Mohit Bansal e Yun-Nung Chen, editores, Anais da Conferência de 2024 sobre Métodos Empíricos em Processamento de Linguagem Natural, EMNLP 2024, Miami, FL, EUA, 12 a 16 de novembro de 2024, páginas 7213–7224. Associação de Linguística Computacional, 2024.
- [115] Tong Liu, Yingjie Zhang, Zhe Zhao, Yinpeng Dong, Guozhu Meng e Kai Chen. Fazendo-os perguntar e responder: Quebrando grandes modelos de linguagem em poucas consultas via disfarce e reconstrução. Em Davide Balzarotti e Wenyan Xu, editores, 33º Simpósio de Segurança da USENIX, USENIX Security 2024, Filadélfia, PA, EUA, 14 a 16 de agosto de 2024. Associação USENIX, 2024.
- [116] Xiao Liu, Liangzhi Li, Tong Xiang, Fuying Ye, Lu Wei, Wangyue Li e Noa Garcia. Imposter.ai: ataques adversários com intenções ocultas em relação a grandes modelos de linguagem alinhados. CoRR, abs/2407.15399, 2024.
- [117] Xiaogeng Liu, Peiran Li, G. Edward Suh, Yevgeniy Vorobeychik, Zhuoqing Mao, Somesh Jha, Patrick McDaniel, Huan Sun, Bo Li e Chaowei Xiao. Autodan-turbo: Um agente vitalício para autoexploração de estratégias para desbloquear llms. In The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapura, 24 a 28 de abril de 2025. OpenReview.net, 2025.
- [118] Xiaogeng Liu, Nan Xu, Muhao Chen e Chaowei Xiao. Autodan: Geração de prompts furtivos de jailbreak em modelos de linguagem grandes alinhados. Em Décima Segunda Conferência Internacional sobre Representações de Aprendizagem, ICLR 2024, Viena, Áustria, 7 a 11 de maio de 2024. OpenReview.net, 2024.
- [119] Xiaoqun Liu, Jiacheng Liang, Muchao Ye e Zhaohan Xi. Robustecendo modelos de linguagem grandes alinhados à segurança por meio de curadoria de dados limpos. CoRR, abs/2405.19358, 2024.
- [120] Xin Liu, Yichen Zhu, Jindong Gu, Yunshi Lan, Chao Yang e Yu Qiao. Mm-safetybench: Um benchmark para avaliação de segurança de grandes modelos de linguagem multimodais. Em Ales Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler e G`ul Varol, editores, Computer Vision - ECCV 2024 - 18ª Conferência Europeia, Milão, Itália, 29 de setembro a 4 de outubro de 2024, Anais, Parte LVI, volume 15114 de Lecture Notes in Computer Science, páginas 386–403. Springer, 2024.

- [121] Yi Liu, Chengjun Cai, Xiaoli Zhang, Xingliang Yuan e Cong Wang. Arondight: Red teaming large vision language models with auto-generated multi-modal jailbreak prompts. In Jianfei Cai, Mohan S. Kankanhalli, Balakrishnan Prab-hakaran, Susanne Boll, Ramanathan Subramanian, Liang Zheng, Vivek K. Singh, Pablo C'esar, Lexing Xie e Dong Xu, editores, Proceedings of the 32nd ACM International Conference on Multimedia, MM 2024, Melbourne, VIC, Australia, 28 October 2024 - 1 November 2024, pages 3578–3586. ACM, 2024.
- [122] Yi Liu, Gelei Deng, Yuekang Li, Kailong Wang, Tianwei Zhang, Yepang Liu, Haoyu Wang, Yan Zheng e Yang Liu. Ataque de injeção imediata contra aplicações integradas ao llm. CoRR, abs/2306.05499, 2023.
- [123] Yi Liu, Gelei Deng, Zhengzi Xu, Yuekang Li, Yaowen Zheng, Ying Zhang, Lida Zhao, Tianwei Zhang e Yang Liu. Desbloqueio do chatgpt via engenharia de prompts: um estudo empírico. CoRR, abs/2305.13860, 2023.
- [124] Yue Liu, Xiaoxin He, Miao Xiong, Jinlan Fu, Shumin Deng e Bryan Hooi. Flipattack: Jailbreak lms via inversão. CoRR, abs/2410.02832, 2024.
- [125] Lin Lu, Hai Yan, Zenghui Yuan, Jiawen Shi, Wenqi Wei, Pin-Yu Chen e Pan Zhou. Autojailbreak: explorando o jailbreak ataques e defesas através da lente da dependência. CoRR, abs/2406.03805, 2024.
- [126] Weikai Lu, Ziqian Zeng, Jianwei Wang, Zhengdong Lu, Zelin Chen, Huiping Zhuang e Cen Chen. Eraser: defesa de jailbreak em grandes modelos de linguagem por meio do desaprendizado de conhecimentos prejudiciais. CoRR, abs/2404.05880, 2024.
- [127] Wen-xiao Lu, Ping Fang, Ming-lu Zhu, Yi-run Zhu, Xinjian Fan, Tian-chen Zhu, Xuan Zhou, Feng-Xia Wang, Tao Chen e Li-ning Sun. Sistema de feedback de reconhecimento de linguagem gestual habilitado para inteligência artificial usando luva inteligente baseada em matrizes de sensores de tensão. Av. Intel. Sistema, 5(8), 2023.
- [128] Weidi Luo, Siyuan Ma, Xiaogeng Liu, Xiaoyu Guo e Chaowei Xiao. Jailbreakv-28k: Um benchmark para avaliar a robustez de modelos de linguagem multimodais grandes contra ataques de jailbreak. CoRR, abs/2404.03027, 2024.
- [129] Huijie Lv, Xiao Wang, Yuansen Zhang, Caishuang Huang, Shihan Dou, Junjie Ye, Tao Gui, Qi Zhang e Xu-anjing Huang. Codechameleon: Estrutura de criptografia personalizada para jailbreak de grandes modelos de linguagem. CoRR, abs/2402.16717, 2024.
- [130] Lijia Lv, Weigang Zhang, Xuehai Tang, Jie Wen, Feng Liu, Jizhong Han e Songlin Hu. Adappa: Abordagem de ataque de jailbreak de pré-preenchimento de posição adaptativa visando llms. CoRR, abs/2409.07503, 2024.
- [131] Jiachen Ma, Yijiang Li, Zhiqing Xiao, Anda Cao, Jie Zhang, Chao Ye e Junbo Zhao. Jailbreaking prompt attack: A controllable adversarial attack against diffusion models. In Luis Chiruzzo, Alan Ritter e Lu Wang, editores, Findings of the Association for Computational Linguistics: NAACL 2025, Albuquerque, New Mexico, EUA, 29 de abril a 4 de maio de 2025, páginas 3141–3157. Association for Computational Linguistics, 2025.
- [132] Jiawei Ma, Po-Yao Huang, Saining Xie, Shang-Wen Li, Luke Zettlemoyer, Shih-Fu Chang, Wen-Tau Yih e Hu Xu. Modo: Especialistas em dados CLIP via agrupamento. Em Conferência IEEE/CVF sobre Visão Computacional e Reconhecimento de Padrões, CVPR 2024, Seattle, WA, EUA, 16 a 22 de junho de 2024, páginas 26344–26353. IEEE, 2024.
- [133] Siyuan Ma, Weidi Luo, Yu Wang, Xiaogeng Liu, Muhao Chen, Bo Li e Chaowei Xiao. Visual-roleplay: ataque universal de jailbreak em modelos de linguagem multimodais grandes via role-playing image character. CoRR, abs/2405.20773, 2024.
- [134] Blazej Manczak, Elliott Zemor, Eric Lin e Vaikkunth Mugunthan. Primeguard: llms seguros e úteis através da sintonia-roteamento livre. CoRR, abs/2407.16318, 2024.
- [135] Neal Mangaokar, Ashish Hooda, Jihye Choi, Shreyas Chandrashekar, Kassem Fawaz, Somesh Jha e Atul Prakash. PRP: propagando perturbações universais para atacar grandes salvaguardas de modelos de linguagem. Em Lun-Wei Ku, Andre Martins e Vivek Srikumar, editores, Anais da 62ª Reunião Anual da Associação de Linguística Computacional (Volume 1: Artigos Longos), ACL 2024, Bangkok, Tailândia, 11 a 16 de agosto de 2024, páginas 10960–10976. Associação de Linguística Computacional, 2024.
- [136] Mantas Mazeika, Long Phan, Xuwang Yin, Andy Zou, Zifan Wang, Norman Mu, Elham Sakhaee, Nathaniel Li, Steven Basart, Bo Li, David A. Forsyth e Dan Hendrycks. Harmbench: Uma estrutura de avaliação padronizada para equipes vermelhas automatizadas e recusa robusta. Em Quadragésima Primeira Conferência Internacional sobre Aprendizado de Máquina, ICML 2024, Viena, Áustria, 21 a 27 de julho de 2024. OpenReview.net, 2024.
- [137] Anay Mehrotra, Manolis Zampetakis, Paul Kassianik, Blaine Nelson, Hyrum S. Anderson, Yaron Singer e Amin Karbasi. Árvore de ataques: quebra automática de llms de caixa preta. Em Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak e Cheng Zhang, editores, Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canadá, 10 a 15 de dezembro de 2024.
- [138] Fredrik Nestaas, Edoardo Debenedetti e Florian Tram`er. Otimização adversarial de mecanismos de busca para grandes modelos de linguagem. Em The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapura, 24 a 28 de abril de 2025. OpenReview.net, 2025.
- [139] Rida Noor, Abdul Wahid, Sibghat Ullah Bazai, Asad Khan, Meie Fang, Syam MS, Uzair Aslam Bhatti e Yazeed Yasin Ghadi. DLGAN: reconstrução de ressonância magnética subamostrada usando rede adversária generativa baseada em aprendizagem profunda. Biomédica. Processamento de sinais. Controle., 93:106218, 2024.
- [140] OpenAI. Relatório técnico GPT-4. CoRR, abs/2303.08774, 2023.
- [141] Dario Pasquini, Martin Strohmaier e Carmela Troncoso. Neural exec: Aprendendo (e aprendendo com) gatilhos de execução para ataques de injeção imediata. Em Maura Pintor, Xinyun Chen e Matthew Jagielski, editores, Anais do Workshop de Inteligência Artificial e Segurança de 2024, AISec 2024, Salt Lake City, UT, EUA, 14 a 18 de outubro de 2024, páginas 89–100. ACM, 2024.

- [142] Anselm Paulus, Arman Zharmagambetov, Chuan Guo, Brandon Amos e Yuandong Tian. Advprompter: Adaptativo rápido estímulo adversário para llms. CoRR, abs/2404.16873, 2024.
- [143] Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley e Jianfeng Gao. Ajuste de instruções com GPT-4. CoRR, abs/2304.03277, 2023.
- [144] Xiangyu Qi, Kaixuan Huang, Ashwinee Panda, Mengdi Wang e Prateek Mittal. Exemplos visuais de adversários desbloqueiam modelos de linguagem de grande porte. CoRR, abs/2306.13213, 2023.
- [145] Cheng Qian, Hainan Zhang, Lei Sha e Zhiming Zheng. HSF: defendendo contra ataques de jailbreak com filtragem de estado oculto. Em Guodong Long, Michale Blumestein, Yi Chang, Liane Lewin-Eytan, Zi Helen Huang e Elad Yom-Tov, editores, Anais Complementares da Conferência ACM sobre a Web 2025, WWW 2025, Sydney, NSW, Austrália, 28 de abril de 2025 - 2 de maio de 2025, páginas 2078–2087. ACM, 2025.
- [146] Huachuan Qiu, Shuai Zhang, Anqi Li, Hongliang He e Zhenzhong Lan. Jailbreak latente: uma referência para avaliação segurança de texto e robustez de saída de grandes modelos de linguagem. CoRR, abs/2307.08487, 2023.
- [147] Md. Abdur Rahman, Hossain Shahriar, Fan Wu e Alfredo Cuzzocrea. Aplicação do BERT multilíngue pré-treinado em embeddings para detecção aprimorada de ataques de injeção de prompts maliciosos. Em 2ª Conferência Internacional sobre Inteligência Artificial, Blockchain e Internet das Coisas, AIBThings 2024, Mt Pleasant, MI, EUA, 7 a 8 de setembro de 2024, páginas 1–7. IEEE, 2024.
- [148] Govind Ramesh, Yao Dou e Wei Xu. GPT-4 se liberta com sucesso quase perfeito usando autoexplicação. Em Yaser Al-Onaizan, Mohit Bansal e Yun-Nung Chen, editores, Anais da Conferência de 2024 sobre Métodos Empíricos em Processamento de Linguagem Natural, EMNLP 2024, Miami, FL, EUA, 12 a 16 de novembro de 2024, páginas 22139–22148. Associação de Linguística Computacional, 2024.
- [149] Delong Ran, Jinyuan Liu, Yichen Gong, Jingyi Zheng, Xinlei He, Tianshuo Cong e Anyu Wang. Jailbreak-eval: Um conjunto de ferramentas integrado para avaliar tentativas de jailbreak contra grandes modelos de linguagem. CoRR, abs/2406.09321, 2024.
- [150] Javier Rando, Francesco Croce, Krystof Mitka, Stepan Shabalin, Maksym Andriushchenko, Nicolas Flammarion e Florian Tram`er. Relatório de competição: Encontrando backdoors universais de jailbreak em llms alinhados. CoRR, abs/2404.14461, 2024.
- [151] Abhinav Rao, Sachin Vashistha, Atharva Naik, Somak Aditya e Monojit Choudhury. Enganando os llms para a desobediência: Compreendendo, analisando e prevenindo invasões de sistemas. CoRR, abs/2305.14965, 2023.
- [152] Bin Ren, Yawei Li, Nancy Mehta, Radu Timofte, Hongyuan Yu, Cheng Wan, Yuxin Hong, Bingnan Han, Zhuoyuan Wu, Yajun Zou, Yuqing Liu, Jizhe Li, Keji He, Chao Fan, Heng Zhang, Xiaolin Zhang, Xuanwu Yin, Kunlong Zuo, Bohao Liao, Peizhe Xia, Long Peng, Zhibo Du, Xin Di, Wangkai Li, Yang Wang, Wei Zhai, Renjing Pei, Jiaming Guo, Songcen Xu, Yang Cao, Zhengjun Zha, Yan Wang, Yi Liu, Qing Wang, Gang Zhang, Liou Zhang, Shijie Zhao, Long Sun, Jinshan Pan, Jiangxin Dong, Jinhui Tang, Xin Liu, Min Yan, Qian Wang, Menghan Zhou, Yiqiang Yan, Yixuan Liu, Wensong Chan, Dehua Tang, Dong Zhou, Li Wang, Lu Tian, Emad Barsoum, Bohan Jia, Junbo Qiao, Yunshuai Zhou, Yun Zhang, Wei Li, Shaohui Lin, Shenglong Zhou, Binbin Chen, Jincheng Liao, Suiyi Zhao, Zhao Zhang, Bo Wang, Yan Luo, Yanyan Wei, Feng Li, Mingshen Wang, Yawei Li, Jinhan Guan, Dehua Hu, Jiawei Yu, Qisheng Xu, Tao Sun, Long Lan, Kele Xu, Xin Lin, Jingtong Yue, Lehan Yang, Shiyi Du, Lu Qi, Chao Ren, Zeyu Han, Yuhang Wang, Chaolin Chen, Haobo Li, Mingjun Zheng, Zhongbao Yang, Lianhong Song, Xingzhuo Yan, Minghan Fu, Jingyi Zhang, Baigang Li, Qi Zhu, Xiaogang Xu, Dan Guo, Chunle Guo, Jiadi Chen, Huanhuan Long, Chunjiang Duanmu, Xiaoyan Lei, Jie Liu, Weilin Jia, Weifeng Cao, Wenlong Zhang, Yanyu Mao, Ruilong Guo, Nihao Zhang, Manoj Pandey, Maksym Chernozhukov, Giang Le, Shuli Cheng, Hongyuan Wang, Ziyang Wei, Qingting Tang, Liejun Wang, Yongming Li, Yanhui Guo, Hao Xu, Akram Khatami-Rizi, Ahmad Mahmoudi-Aznaveh, Chih-Chung Hsu, Chia-Ming Lee, Yi-Shiuan Chou, Amogh Joshi, Nikhil Akalwadi, Sampada Malagi, Palani Yashaswini, Chaitra Desai, Ramesh Ashok Tabib, Ujwala Patil e Uma Mudanagudi. O nono relatório do desafio de super-resolução eficiente NTIRE 2024. Na Conferência IEEE/CVF sobre Visão Computacional e Reconhecimento de Padrões, CVPR 2024 - Workshops, Seattle, WA, EUA, 17 a 18 de junho de 2024, páginas 6595–6631. IEEE, 2024.
- [153] Qibing Ren, Chang Gao, Jing Shao, Junchi Yan, Xin Tan, Wai Lam e Lizhuang Ma. Codeattack: Revelando desafios de generalização de segurança de grandes modelos de linguagem por meio de preenchimento de código. Em Lun-Wei Ku, Andre Martins e Vivek Srikumar, editores, Anais da Associação de Linguística Computacional, ACL 2024, Bangkok, Tailândia e reunião virtual, 11 a 16 de agosto de 2024, páginas 11437–11452. Associação de Linguística Computacional, 2024.
- [154] Qibing Ren, Hao Li, Dongrui Liu, Zhanxu Xie, Xiaoya Lu, Yu Qiao, Lei Sha, Junchi Yan, Lizhuang Ma e Jing Shao. Descarrile-se: Ataque de fuga da prisão LLM multi-turno por meio de pistas descobertas por você mesmo. CoRR, abs/2410.10700, 2024.
- [155] Yupeng Ren. F2A: uma abordagem inovadora para injeção rápida utilizando agentes de detecção de segurança simulada. CoRR, abs/2410.08776, 2024.
- [156] Alexander Robey, Eric Wong, Hamed Hassani e George J. Pappas. Smoothllm: Defendendo grandes modelos de linguagem contra ataques de jailbreak. Trans. Mach. Learn. Res., 2025, 2025.
- [157] Mark Russinovich, Ahmed Salem e Ronen Eldan. Ótimo, agora escreva um artigo sobre isso: O ataque de fuga LLM de múltiplas voltas em crescendo. CoRR, abs/2404.01833, 2024.
- [158] Sander Schulhoff, Jeremy Pinto, Ansum Khan, Louis-François Bouchard, Chenglei Si, Svetlana Anati, Valen Tagliabue, Anson Liu Kost, Christopher Carnahan e Jordan L. Boyd-Graber. Ignore este título e hackaprompt: Expondo vulnerabilidades sistêmicas de llms por meio de uma competição global de hacking rápido. Em Houda Bouamor, Juan Pino e Kalika Bali, editores, Anais da Conferência de 2023 sobre Métodos Empíricos em Processamento de Linguagem Natural, EMNLP 2023, Cingapura, 6 a 10 de dezembro de 2023, páginas 4945–4977. Associação para Linguística Computacional, 2023.

- [159] Reshabh K. Sharma, Vinayak Gupta e Dan Grossman. SPML: Uma DSL para defender modelos de linguagem contra prompts ataques. CoRR, abs/2402.11755, 2024.
- [160] Erfan Shayegani, Md Abdullah Al Mamun, Yu Fu, Pedram Zaree, Yue Dong e Nael B. Abu-Ghazaleh. Levantamento de vulnerabilidades em grandes modelos de linguagem reveladas por ataques adversários. CoRR, abs/2310.10844, 2023.
- [161] Guobin Shen, Dongcheng Zhao, Yiting Dong, Xiang He e Yi Zeng. Antídoto para jailbreak: equilíbrio segurança-utilidade em tempo de execução via ajuste de representação esparsa em grandes modelos de linguagem. CoRR, abs/2410.02298, 2024.
- [162] Guobin Shen, Dongcheng Zhao, Yiting Dong, Xiang He e Yi Zeng. Antídoto para o jailbreak: equilíbrio entre segurança e utilidade em tempo de execução por meio do ajuste de representação esparsa em grandes modelos de linguagem. In Décima Terceira Conferência Internacional sobre Representações de Aprendizagem, ICLR 2025, Singapura, 24 a 28 de abril de 2025. OpenReview.net, 2025.
- [163] Xinyue Shen, Zeyuan Chen, Michael Backes, Yun Shen e Yang Zhang. "faça qualquer coisa agora": caracterizando e avaliando prompts de jailbreak em modelos de linguagem de grande porte. Em Bo Luo, Xiaojing Liao, Jun Xu, Engin Kirda e David Lie, editores, Anais da Conferência SIGSAC de 2024 da ACM sobre Segurança de Computadores e Comunicações, CCS 2024, Salt Lake City, UT, EUA, 14 a 18 de outubro de 2024, páginas 1671–1685. ACM, 2024.
- [164] Jiawen Shi, Zenghui Yuan, YINUO Liu, Yue Huang, Pan Zhou, Lichao Sun e Neil Zhenqiang Gong. Ataque de injeção rápida baseado em otimização para llm-como-juiz. Em Bo Luo, Xiaojing Liao, Jun Xu, Engin Kirda e David Lie, editores, Anais da Conferência SIGSAC de 2024 da ACM sobre Segurança de Computadores e Comunicações, CCS 2024, Salt Lake City, UT, EUA, 14 a 18 de outubro de 2024, páginas 660–674. ACM, 2024.
- [165] Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace e Sameer Singh. Autoprompt: Obtendo conhecimento a partir de modelos de linguagem com prompts gerados automaticamente. Em Bonnie Webber, Trevor Cohn, Yulan He e Yang Liu, editores, Anais da Conferência de 2020 sobre Métodos Empíricos em Processamento de Linguagem Natural, EMNLP 2020, Online, 16 a 20 de novembro de 2020, páginas 4222–4235. Associação para Linguística Computacional, 2020.
- [166] Sonali Singh, Faranak Abri e Akbar Siami Namin. Explorando grandes modelos de linguagem (LLMs) por meio de técnicas de engano e princípios de persuasão. Em Jingrui He, Themis Palpanas, Xiaohua Hu, Alfredo Cuzzocrea, Dejing Dou, Dominik Slezak, Wei Wang, Aleksandra Gruca, Jerry Chun-Wei Lin e Rakesh Agrawal, editores, IEEE International Conference on Big Data, BigData 2023, Sorrento, Itália, 15 a 18 de dezembro de 2023, páginas 2508–2517. IEEE, 2023.
- [167] Chawin Sitawarin, Norman Mu, David A. Wagner e Alexandre Araujo. PAL: ataque de caixa preta guiado por proxy em grandes modelos de linguagem. CoRR, abs/2402.09674, 2024.
- [168] Alexandra Souly, Qingyuan Lu, Dillon Bowen, Tu Trinh, Elvis Hsieh, Sana Pandey, Pieter Abbeel, Justin Svegliato, Scott Emmons, Olivia Watkins e Sam Toyer. Uma forte rejeição para fugas de presos vazias. Em Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak e Cheng Zhang, editores, Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canadá, 10 a 15 de dezembro de 2024.
- [169] Gianluca De Stefano, Lea Schönherr e Giancarlo Pellegrino. Rag and roll: Uma avaliação ponta a ponta do prompt indireto manipulações em frameworks de aplicações baseadas em llm. CoRR, abs/2408.05025, 2024.
- [170] Penghao Sun, Julong Lan, Yuxiang Hu, Zehua Guo, Chong Wu e Jiangxing Wu. Realizando o serviço consciente do carbono provisão no sistema de TIC. IEEE Trans. Netw. Serv. Manag., 21(4):4090–4103, 2024.
- [171] Zhifan Sun e Antonio Valerio Miceli Barone. Comportamento de escalabilidade da tradução automática com grandes modelos de linguagem sob ataques de injeção de prompts. CoRR, abs/2403.09832, 2024.
- [172] Equipe ByteDance Seed. Relatório técnico Seed1.5-vl. arXiv preprint arXiv:2505.07062, 2025.
- [173] Teknium. Openhermes 2.5: Um conjunto de dados aberto de dados sintéticos para assistentes generalistas de LLM, 2023. Disponível em: <https://huggingface.co/datasets/teknium/OpenHermes-2.5>.
- [174] Yu Tian, Xiao Yang, Jingyuan Zhang, Yinpeng Dong e Hang Su. Gênios do mal: investigando a segurança dos filmes baseados em agentes. CoRR, abs/2311.11855, 2023.
- [175] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Roziere, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave e Guillaume Lample. Llama: Modelos de linguagem básicos abertos e eficientes. CoRR, abs/2302.13971, 2023.
- [176] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajwal Bhargava, Shrutu Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov e Thomas Scialom. Llama 2: Base aberta e modelos de chat ajustados. CoRR, abs/2307.09288, 2023.
- [177] Sam Toyer, Olivia Watkins, Ethan Adrian Mendes, Justin Svegliato, Luke Bailey, Tiffany Wang, Isaac Ong, Karim Elmaaroufi, Pieter Abbeel, Trevor Darrell, Alan Ritter e Stuart Russell. Confiança de tensores: ataques de injeção de prompts interpretáveis a partir de um jogo online. In Décima Segunda Conferência Internacional sobre Representações de Aprendizagem, ICLR 2024, Viena, Áustria, 7 a 11 de maio de 2024. OpenReview.net, 2024.

- [178] Shangqing Tu, Zhuoran Pan, Wenxuan Wang, Zhixin Zhang, Yuliang Sun, Jifan Yu, Hongning Wang, Lei Hou e Juanzi Li. Do conhecimento ao jailbreak: Um ponto de conhecimento vale um ataque. CoRR, abs/2406.11682, 2024.
- [179] Dmitrii Volkov. Badllama 3: removendo o ajuste fino de segurança do llama 3 em minutos. CoRR, abs/2407.01376, 2024.
- [180] Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner e Sameer Singh. Gatilhos adversários universais para atacar e analisar PNL. Em Kentaro Inui, Jing Jiang, Vincent Ng e Xiaojun Wan, editores, Anais da Conferência de 2019 sobre Métodos Empíricos em Processamento de Linguagem Natural e da 9ª Conferência Internacional Conjunta sobre Processamento de Linguagem Natural, EMNLP-IJCNLP 2019, Hong Kong, China, 3 a 7 de novembro de 2019, páginas 2153–2162. Associação de Linguística Computacional, 2019.
- [181] Eric Wallace, Kai Xiao, Reimar Leike, Lilian Weng, Johannes Heidecke e Alex Beutel. A hierarquia de instruções: Treinamento de LLMs para priorizar instruções privilegiadas. CoRR, abs/2404.13208, 2024.
- [182] Hao Wang, Hao Li, Minlie Huang e Lei Sha. Do ruído à clareza: Desvendando o sufixo adversarial da linguagem ampla ataques de modelos via tradução de embeddings de texto. CoRR, abs/2402.16006, 2024.
- [183] Haoyu Wang, Bingzhe Wu, Yatao Bian, Yongzhe Chang, Xueqian Wang e Peilin Zhao. Sondando o limite de resposta de segurança de grandes modelos de linguagem por meio da geração de caminhos de decodificação inseguros. CoRR, abs/2408.10668, 2024.
- [184] Jiong Xiao Wang, Jiazhao Li, Yiquan Li, Xiangyu Qi, Junjie Hu, Yixuan Li, Patrick McDaniel, Muhao Chen, Bo Li e Chaowei Xiao. Mitigando o ataque de jailbreak de ajuste fino com alinhamento aprimorado de backdoor. CoRR, abs/2402.14968, 2024.
- [185] Jiong Xiao Wang, Zichen Liu, Keun Hee Park, Muhao Chen e Chaowei Xiao. Ataques de demonstração adversária em grandes modelos de linguagem. CoRR, abs/2305.14950, 2023.
- [186] Mengru Wang, Ningyu Zhang, Ziwen Xu, Zekun Xi, Shumin Deng, Yunzhi Yao, Qishen Zhang, Linyi Yang, Jindong Wang e Huajun Chen. Desintoxicação de grandes modelos de linguagem por meio de edição de conhecimento. Em Lun-Wei Ku, Andre Martins e Vivek Srikumar, editores, Anais da 62ª Reunião Anual da Associação de Linguística Computacional (Volume 1: Artigos Longos), ACL 2024, Bangkok, Tailândia, 11 a 16 de agosto de 2024, páginas 3093–3118. Associação de Linguística Computacional, 2024.
- [187] Peiran Wang, Xiaogeng Liu e Chaowei Xiao. Repd: Defesa contra ataques de jailbreak por meio de um processo de decomposição de prompts baseado em recuperação. CoRR, abs/2410.08660, 2024.
- [188] Peiran Wang, Xiaogeng Liu e Chaowei Xiao. Repd: Defending jailbreak attack through a retrieval-based prompt decomposition process. In Luis Chiruzzo, Alan Ritter e Lu Wang, editores, Findings of the Association for Computational Linguistics: NAACL 2025, Albuquerque, New Mexico, USA, April 29 - May 4, 2025, pages 283–294. Association for Computational Linguistics, 2025.
- [189] Wenxuan Wang, Kuiyi Gao, Zihan Jia, Youliang Yuan, Jen-tse Huang, Qiuzhi Liu, Shuai Wang, Wenxiang Jiao e Zhaopeng Tu. Ataque de cadeia de jailbreak para modelos de geração de imagens via edição passo a passo. CoRR, abs/2410.03869, 2024.
- [190] Xinyuan Wang, Victor Shea-Jay Huang, Renmiao Chen, Hao Wang, Chengwei Pan, Lei Sha e Minlie Huang. Black-dan: Uma abordagem multi-objetivo de caixa preta para o jailbreaking eficaz e contextual de grandes modelos de linguagem. CoRR, abs/2410.09804, 2024.
- [191] Xuguang Wang, Daoyuan Wu, Zhenlan Ji, Zongjie Li, Pingchuan Ma, Shuai Wang, Yingjiu Li, Yang Liu, Ning Liu e Juergen Rahmel. Autodefesa: Lms podem se defender contra o jailbreak de maneira prática. CoRR, abs/2406.05498, 2024.
- [192] Yihan Wang, Zhouxing Shi, Andrew Bai e Cho-Jui Hsieh. Defendendo lms contra ataques de jailbreaking via retrotradução. Em Lun-Wei Ku, Andre Martins e Vivek Srikumar, editores, Anais da Associação de Linguística Computacional, ACL 2024, Bangkok, Tailândia e reunião virtual, 11 a 16 de agosto de 2024, páginas 16031–16046. Associação de Linguística Computacional, 2024.
- [193] Yu Wang, Xiaogeng Liu, Yu Li, Muhao Chen e Chaowei Xiao. Adashield: Protegendo modelos de linguagem multimodais grandes contra ataques baseados em estrutura por meio de prompts de escudo adaptativos. CoRR, abs/2403.09513, 2024.
- [194] Zi Wang, Divyam Anshuman, Ashish Hooda, Yudong Chen e Somesh Jha. Homotopia funcional: suavização da otimização discreta por meio de parâmetros contínuos para ataques de jailbreak LLM. Na Décima Terceira Conferência Internacional sobre Representações de Aprendizagem, ICLR 2025, Singapura, 24 a 28 de abril de 2025. OpenReview.net, 2025.
- [195] Ziqiu Wang, Jun Liu, Shengkai Zhang e Yang Yang. Longchain envenenado: Jailbreak lms por longchain. CoRR, abs/2406.18122, 2024.
- [196] Alexander Wei, Nika Haghtalab e Jacob Steinhardt. Jailbroken: Como o treinamento de segurança LLM falha? Em Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt e Sergey Levine, editores, Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, Nova Orleans, LA, EUA, 10 a 16 de dezembro de 2023.
- [197] Zeming Wei, Yifei Wang e Yisen Wang. Modelos de linguagem alinhados por jailbreak e guard com apenas alguns contextos. demonstrações. CoRR, abs/2310.06387, 2023.
- [198] Fenghua Weng, Yue Xu, Chengyan Fu e Wenjie Wang. Mmj-bench: Um estudo abrangente sobre ataques e defesas de jailbreak para modelos de linguagem de visão. Em Toby Walsh, Julie Shah e Zico Kolter, editores, AAAI-25, Patrocinado pela Associação para o Avanço da Inteligência Artificial, 25 de fevereiro a 4 de março de 2025, Filadélfia, PA, EUA, páginas 27689–27697. AAAI Press, 2025.

- [199] WizardLMTeam. Wizardlm evol instruct 70k, 2023. Disponível em: https://huggingface.co/datasets/WizardLMTeam/WizardLM_evol_instruct_70k.
- [200] Yuanwei Wu, Xiang Li, Yixin Liu, Pan Zhou e Lichao Sun. Jailbreak GPT-4V por meio de ataques auto-adversários com avisos do sistema. CoRR, abs/2311.09127, 2023.
- [201] Zeguan Xiao, Yan Yang, Guanhua Chen e Yun Chen. Tastle: Distrair grandes modelos de linguagem para jailbreak automático ataque. CoRR, abs/2403.08424, 2024.
- [202] Yueqi Xie, Minghong Fang, Renjie Pi e Neil Zhenqiang Gong. Gradsafe: Detectando prompts inseguros para llms via Análise de gradiente crítica para a segurança. CoRR, abs/2402.13494, 2024.
- [203] Chen Xiong, Xiangyu Qi, Pin-Yu Chen e Tsung-Yi Ho. Patch de alerta defensivo: uma defesa robusta e interpretável de llms contra ataques de jailbreak. CoRR, abs/2405.20099, 2024.
- [204] Huiyu Xu, Wenhui Zhang, Zhibo Wang, Feng Xiao, Rui Zheng, Yunhe Feng, Zhongjie Ba e Kui Ren. Redagente: Vermelho Combinando grandes modelos de linguagem com um agente de linguagem autônomo sensível ao contexto. CoRR, abs/2407.16667, 2024.
- [205] Rongwu Xu, Zehan Qi e Wei Xu. "Ataques" de resposta preventiva ao raciocínio em cadeia de pensamento. Em Lun-Wei Ku, Andre Martins e Vivek Srikumar, editores, Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Tailândia e reunião virtual, 11 a 16 de agosto de 2024, páginas 14708–14726. Association for Computational Linguistics, 2024.
- [206] Zhangchen Xu, Fengqing Jiang, Luyao Niu, Jinyuan Jia, Bill Yuchen Lin e Radha Poovendran. Safedecoding: Defendendo-se contra ataques de jailbreak por meio de decodificação com reconhecimento de segurança. Em Lun-Wei Ku, Andre Martins e Vivek Srikumar, editores, Anais da 62ª Reunião Anual da Associação de Linguística Computacional (Volume 1: Artigos Longos), ACL 2024, Bangkok, Tailândia, 11 a 16 de agosto de 2024, páginas 5587–5605. Associação de Linguística Computacional, 2024.
- [207] Zhao Xu, Fan Liu e Hao Liu. Bag of tricks: Benchmarking of jailbreak attacks on llms. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak e Cheng Zhang, editores, Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canadá, 10 a 15 de dezembro de 2024.
- [208] Zihao Xu, Yi Liu, Gelei Deng, Yuekang Li e Stjepan Picek. Um estudo abrangente de ataque versus defesa de jailbreak para grandes modelos de linguagem. Em Lun-Wei Ku, Andre Martins e Vivek Srikumar, editores, Anais da Associação de Linguística Computacional, ACL 2024, Bangkok, Tailândia e reunião virtual, 11 a 16 de agosto de 2024, páginas 7432–7449. Associação para Linguística Computacional, 2024.
- [209] An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jianxin Yang, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Xuejing Liu, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yeqiong Liu, Zeyu Cui, Zhenru Zhang, Zhifang Guo e Zhihao Fan. Relatório técnico Qwen2. CoRR, abs/2407.10671, 2024.
- [210] An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yeqiong Liu, Zeyu Cui, Zhenru Zhang e Zihan Qiu. Relatório técnico Qwen2.5. CoRR, abs/2412.15115, 2024.
- [211] Sin-Han Yang, Tuomas P. Oikarinen e Tsui-Wei Weng. Aprendizagem contínua orientada por conceitos. Trans. Mach. Learn. Res., 2024, 2024.
- [212] Xikang Yang, Xuehai Tang, Songlin Hu e Jizhong Han. Chain of attack: a semantic-driven contextual multi-turn attacker for LLM. CoRR, abs/2405.05610, 2024.
- [213] Yan Yang, Zeguan Xiao, Xin Lu, Hongru Wang, Hailiang Huang, Guanhua Chen e Yun Chen. Sop: Desbloqueie o poder de facilitação social para ataque de jailbreak automático. CoRR, abs/2407.01902, 2024.
- [214] Yuchen Yang, Bo Hui, Haolin Yuan, Neil Gong e Yinzi Cao. Sneakyprompt: Jailbreaking text-to-image generative models. In IEEE Symposium on Security and Privacy, SP 2024, San Francisco, CA, USA, May 19-23, 2024, pages 897–912. IEEE, 2024.
- [215] Dongyu Yao, Jianshu Zhang, Ian G. Harris e Marcel Carlsson. Fuzzllm: Uma estrutura de fuzzing inovadora e universal para descobrir proativamente vulnerabilidades de jailbreak em grandes modelos de linguagem. CoRR, abs/2309.05274, 2023.
- [216] Junjie Ye, Sixian Li, Guanyu Li, Caishuang Huang, Songyang Gao, Yilong Wu, Qi Zhang, Tao Gui e Xuanjing Huang. Toolsword: Revelando questões de segurança de grandes modelos de linguagem na aprendizagem de ferramentas em três estágios. Em Lun-Wei Ku, Andre Martins e Vivek Srikumar, editores, Anais da 62ª Reunião Anual da Associação de Linguística Computacional (Volume 1: Artigos Longos), ACL 2024, Bangkok, Tailândia, 11 a 16 de agosto de 2024, páginas 2181–2211. Associação de Linguística Computacional, 2024.
- [217] Siboyi, Yule Liu, Zhen Sun, Tianshuo Cong, Xinlei He, Jiaying Song, Ke Xu e Qi Li. Ataques e defesas de jailbreak contra grandes modelos de linguagem: Uma pesquisa. CoRR, abs/2407.04295, 2024.
- [218] Zonghao Ying, Aishan Liu, Tianyuan Zhang, Zhengmin Yu, Siyuan Liang, Xianglong Liu e Dacheng Tao. Modelos de linguagem de visão de jailbreak por meio de prompt adversário bimodal. CoRR, abs/2406.04031, 2024.

- [219] Zheng Xin Yong, Cristina Menghini e Stephen H. Bach. Linguagens de poucos recursos fazem jailbreak do GPT-4. CoRR, abs/2310.02446, 2023.
- [220] Jiahao Yu, Xingwei Lin, Zheng Yu e Xinyu Xing. GPTFUZZER: red unindo grandes modelos de linguagem com auto-prompts de jailbreak gerados. CoRR, abs/2309.10253, 2023.
- [221] Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengyong Liu, Yu Zhang, James T. Kwok, Zhenguo Li, Adrian Weller e Weiyang Liu. Metamath: Inicialize suas próprias perguntas matemáticas para grandes modelos de linguagem. Em Décima Segunda Conferência Internacional sobre Representações de Aprendizagem, ICLR 2024, Viena, Áustria, 7 a 11 de maio de 2024. OpenReview.net, 2024.
- [222] Zhiyuan Yu, Xiaogeng Liu, Shunning Liang, Zach Cameron, Chaowei Xiao e Ning Zhang. Não me ouçam: Compreendendo e explorando prompts de jailbreak de grandes modelos de linguagem. Em Davide Balzarotti e Wenyuan Xu, editores, 33º Simpósio de Segurança da USENIX, USENIX Security 2024, Filadélfia, PA, EUA, 14 a 16 de agosto de 2024. USENIX Association, 2024.
- [223] Youliang Yuan, Wenxiang Jiao, Wenxuan Wang, Jen-tse Huang, Pinjia He, Shuming Shi e Zhaopeng Tu. GPT-4 é inteligente demais para ser seguro: bate-papo furtivo com llms via cifra. Em Décima Segunda Conferência Internacional sobre Representações de Aprendizagem, ICLR 2024, Viena, Áustria, 7 a 11 de maio de 2024. OpenReview.net, 2024.
- [224] Zhuowen Yuan, Zidi Xiong, Yi Zeng, Ning Yu, Ruoxi Jia, Dawn Song e Bo Li. Rigorllm: Guardrails resilientes para grandes modelos de linguagem contra conteúdo indesejado. Em Quadragésima Primeira Conferência Internacional sobre Aprendizado de Máquina, ICML 2024, Viena, Áustria, 21 a 27 de julho de 2024. OpenReview.net, 2024.
- [225] Xinyi Zeng, Yuying Shang, Yutao Zhu, Jiawei Chen e Yu Tian. Estratégias de defesa raiz: Garantindo a segurança do LLM no nível de decodificação. CoRR, abs/2410.06809, 2024.
- [226] Yi Zeng, Hongpeng Lin, Jingwen Zhang, Diyi Yang, Ruoxi Jia e Weiyan Shi. Como Johnny pode persuadir os LLMs a desbloqueá-los: repensando a persuasão para desafiar a segurança da IA humanizando os LLMs. Em Lun-Wei Ku, Andre Martins e Vivek Srikumar, editores, Anais da 62ª Reunião Anual da Associação de Linguística Computacional (Volume 1: Artigos Longos), ACL 2024, Bangkok, Tailândia, 11 a 16 de agosto de 2024, páginas 14322–14350. Associação de Linguística Computacional, 2024.
- [227] Yi Zeng, Yu Yang, Andy Zhou, Jeffrey Ziwei Tan, Yuheng Tu, Yifan Mai, Kevin Klyman, Minzhou Pan, Ruoxi Jia, Dawn Song, Percy Liang e Bo Li. Air-bench 2024: Uma referência de segurança baseada em categorias de risco de regulamentos e políticas. CoRR, abs/2407.17436, 2024.
- [228] Yifan Zeng, Yiran Wu, Xiao Zhang, Huazheng Wang e Qingyun Wu. Autodefesa: Defesa LLM multiagente contra ataques de jailbreak. CoRR, abs/2403.04783, 2024.
- [229] Qiusi Zhan, Zhixiang Liang, Zifan Ying e Daniel Kang. Injecagent: Benchmarking indirect prompt injections in tool-integrated large language model agents. In Lun-Wei Ku, Andre Martins e Vivek Srikumar, editores, Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Tailândia e reunião virtual, 11 a 16 de agosto de 2024, páginas 10471–10506. Association for Computational Linguistics, 2024.
- [230] Chong Zhang, Mingyu Jin, Qinkai Yu, Chengzhi Liu, Haochen Xue e Xiaobo Jin. Ataque de injeção de prompt generativo guiado por objetivo em grandes modelos de linguagem. Em Elena Baralis, Kun Zhang, Ernesto Damiani, Mérouane Debbah, Panos Kalnis e Xindong Wu, editores, IEEE International Conference on Data Mining, ICDM 2024, Abu Dhabi, Emirados Árabes Unidos, 9 a 12 de dezembro de 2024, páginas 941–946. IEEE, 2024.
- [231] Hangfan Zhang, Zhimeng Guo, Huaisheng Zhu, Bochuan Cao, Lu Lin, Jinyuan Jia, Jinghui Chen e Dinghao Wu. Modelos de linguagem de código aberto de grande porte que escapam da prisão por meio de decodificação forçada. Em Lun-Wei Ku, Andre Martins e Vivek Srikumar, editores, Anais da 62ª Reunião Anual da Associação de Linguística Computacional (Volume 1: Artigos Longos), ACL 2024, Bangkok, Tailândia, 11 a 16 de agosto de 2024, páginas 5475–5493. Associação de Linguística Computacional, 2024.
- [232] Xiaoyu Zhang, Cen Zhang, Tianlin Li, Yihao Huang, Xiaojun Jia, Xiaofei Xie, Yang Liu e Chao Shen. Uma mutação-Método baseado em multimodalidade para detecção de ataques de jailbreak. CoRR, abs/2312.10766, 2023.
- [233] Yuqi Zhang, Liang Ding, Lefei Zhang e Dacheng Tao. O prompt de análise de intenção torna grandes modelos de linguagem A bom defensor de jailbreak. CoRR, abs/2401.06561, 2024.
- [234] Zaibin Zhang, Yongting Zhang, Lijun Li, Jing Shao, Hongzhi Gao, Yu Qiao, Lijun Wang, Huchuan Lu e Feng Zhao. Psysafe: Uma estrutura abrangente para ataque, defesa e avaliação da segurança de sistemas multiagentes com base em princípios psicológicos. Em Lun-Wei Ku, Andre Martins e Vivek Srikumar, editores, Anais da 62ª Reunião Anual da Associação de Linguística Computacional (Volume 1: Artigos Longos), ACL 2024, Bangkok, Tailândia, 11 a 16 de agosto de 2024, páginas 15202–15231. Associação de Linguística Computacional, 2024.
- [235] Zhixin Zhang, Junxiao Yang, Pei Ke, Shiyao Cui, Chujie Zheng, Hongning Wang e Minlie Huang. Desaprendizagem segura: uma solução surpreendentemente eficaz e generalizável para defesa contra ataques de jailbreak. CoRR, abs/2407.02855, 2024.
- [236] Zhuo Zhang, Guangyu Shen, Guanhong Tao, Siyuan Cheng e Xiangyu Zhang. Sobre a resiliência de modelos de linguagem de grande porte a interrogatórios coercitivos. Em IEEE Symposium on Security and Privacy, SP 2024, São Francisco, CA, EUA, 19 a 23 de maio de 2024, páginas 826–844. IEEE, 2024.
- [237] Bowen Zhao, Wei-Neng Chen, Feng-Feng Wei, Ximeng Liu, Qingqi Pei e Jun Zhang. PEGA: Um algoritmo genético que preserva a privacidade para otimização combinatória. IEEE Trans. Cybern., 54(6):3638–3651, 2024.
- [238] Jiawei Zhao, Kejiang Chen, Xiaojian Yuan e Weiming Zhang. Orientação de prefixo: um volante para linguagem extensa Modelos para defesa contra ataques de jailbreak. CoRR, abs/2408.08924, 2024.

- [239] Wei Zhao, Zhe Li, Yige Li e Jun Sun. Sufixos adversários também podem ser recursos! CoRR, abs/2410.00451, 2024.
- [240] Wei Zhao, Zhe Li, Yige Li, Ye Zhang e Jun Sun. Defesa de grandes modelos de linguagem contra ataques de jailbreak por meio de edição específica de camada. Em Yaser Al-Onaizan, Mohit Bansal e Yun-Nung Chen, editores, Findings of the Association for Computational Linguistics: EMNLP 2024, Miami, Flórida, EUA, 12 a 16 de novembro de 2024, páginas 5094–5109. Associação de Linguística Computacional,
- [241] Xuandong Zhao, Xianjun Yang, Tianyu Pang, Chao Du, Lei Li, Yu-Xiang Wang e William Yang Wang. Fraco para forte jailbreaking em grandes modelos de linguagem. CoRR, abs/2401.17256, 2024.
- [242] Chujie Zheng, Fan Yin, Hao Zhou, Fandong Meng, Jie Zhou, Kai-Wei Chang, Minlie Huang e Nanyun Peng. Proteção LLM orientada por prompt por meio de otimização de representação direcionada. CoRR, abs/2401.18018, 2024.
- [243] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Tianle Li, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zhuohan Li, Zì Lin, Eric P. Xing, Joseph E. Gonzalez, Ion Stoica e Hao Zhang. Lmsys-chat-1m: Um conjunto de dados de conversação LLM do mundo real em grande escala. Na Décima Segunda Conferência Internacional sobre Representações de Aprendizagem, ICLR 2024, Viena, Áustria, 7 a 11 de maio de 2024. OpenReview.net, 2024.
- [244] Xiaosen Zheng, Tianyu Pang, Chao Du, Qian Liu, Jing Jiang e Min Lin. O aprimoramento do jailbreaking com poucos exemplos pode contornar modelos de linguagem alinhados e suas defesas. Em Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak e Cheng Zhang, editores, Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canadá, 10 a 15 de dezembro de 2024.
- [245] Andy Zhou, Bo Li e Haohan Wang. Otimização robusta de prompts para defesa de modelos de linguagem contra ataques de jailbreak. Em Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak e Cheng Zhang, editores, Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canadá, 10 a 15 de dezembro de 2024.
- [246] Yukai Zhou e Wenjie Wang. Não diga não: Quebrando o LLM pela supressão da recusa. CoRR, abs/2404.16369, 2024.
- [247] Yuqi Zhou, Lin Lu, Ryan Sun, Pan Zhou e Lichao Sun. Ataques de jailbreak com aprimoramento de contexto virtual e injeção de token especial. Em Yaser Al-Onaizan, Mohit Bansal e Yun-Nung Chen, editores, Findings of the Association for Computational Linguistics: EMNLP 2024, Miami, Flórida, EUA, 12 a 16 de novembro de 2024, páginas 11843–11857. Association for Computational Linguistics, 2024.
- [248] Sicheng Zhu, Ruiyi Zhang, Bang An, Gang Wu, Joe Barrow, Zichao Wang, Furong Huang, Ani Nenkova e Tong Sun. Autodan: ataques adversários interpretáveis baseados em gradiente em grandes modelos de linguagem. arXiv preprint arXiv:2310.15140, 2023.
- [249] Yongshuo Zong, Ondrej Bohdal, Tingyang Yu, Yongxin Yang e Timothy M. Hospedales. Ajuste fino de segurança a (quase) nenhum custo: Uma linha de base para grandes modelos de linguagem de visão. Em Quadragésima Primeira Conferência Internacional sobre Aprendizado de Máquina, ICML 2024, Viena, Áustria, 21 a 27 de julho de 2024. OpenReview.net, 2024.
- [250] Andy Zou, Long Phan, Justin Wang, Derek Duenas, Maxwell Lin, Maksym Andriushchenko, J. Zico Kolter, Matt Fredrikson e Dan Hendrycks. Melhorando o alinhamento e a robustez com disjuntores. Em Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak e Cheng Zhang, editores, Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canadá, 10 a 15 de dezembro de 2024.
- [251] Andy Zou, Zifan Wang, J. Zico Kolter e Matt Fredrikson. Ataques adversários universais e transferíveis em modelos de linguagem alinhados. CoRR, abs/2307.15043, 2023.
- [252] Xiaotian Zou e Yongkang Chen. Quebra da lógica de imagem para texto: Sua imaginação pode te ajudar a fazer qualquer coisa. CoRR, abs/2407.02534, 2024.
- [253] Xiaotian Zou, Yongkang Chen e Ke Li. A mensagem do sistema é realmente importante para jailbreaks em grandes modelos de linguagem? CoRR, abs/2402.14857, 2024.