GeneBreaker: Jailbreak Attacks against DNA Language Models with Pathogenicity Guidance

Zaixi Zhang[*,†]
Princeton University
zz8680@princeton.edu

Zhenghong Zhou[*]
Shanghai Jiao Tong University
lltzahd615@sjtu.edu.cn

Ruofan Jin[*,‡]
Zhejiang University
ruofanjin@zju.edu.cn

Le Cong[†]
Stanford University
congle@stanford.edu

Mengdi Wang[†]
Princeton University
mengdiw@princeton.edu

Abstract

DNA, encoding genetic instructions for almost all living organisms, fuels groundbreaking advances in genomics and synthetic biology. Recently, DNA Foundation Models have achieved success in designing synthetic functional DNA sequences, even whole genomes, but their susceptibility to jailbreaking remains underexplored, leading to potential concern of generating harmful sequences such as pathogens or toxin-producing genes. In this paper, we introduce GeneBreaker, the first framework to systematically evaluate jailbreak vulnerabilities of DNA foundation models. GeneBreaker employs (1) an LLM agent with customized bioinformatic tools to design high-homology, non-pathogenic jailbreaking prompts, (2) beam search guided by PathoLM and log-probability heuristics to steer generation toward pathogen-like sequences, and (3) a BLAST-based evaluation pipeline against a curated Human Pathogen Database (JailbreakDNABench) to detect successful jailbreaks. Evaluated on our JailbreakDNABench, GeneBreaker successfully jailbreaks the latest Evo series models across 6 viral categories consistently (up to 60% Attack Success Rate for Evo2-40B). Further case studies on SARS-CoV-2 spike protein and HIV-1 envelope protein demonstrate the sequence and structural fidelity of jailbreak output, while evolutionary modeling of SARS-CoV-2 underscores biosecurity risks. Our findings also reveal that scaling DNA foundation models amplifies dual-use risks, motivating enhanced safety alignment and tracing mechanisms. Our code is at `https://github.com/zaixizhang/GeneBreaker`.

Disclaimer: This paper contains potentially offensive and harmful content.

1 Introduction

DNA, as the fundamental blueprint of life, underpins biological processes and holds immense potential for advancing genomics and synthetic biology [15, 58, 9]. Recently, DNA foundation models, such as DNABert [27, 81], Nucleotide Transformer[16], Generator[67], and Evo series [39, 11], have transformed genomics by enabling unprecedented capabilities in sequence generation and analysis. However, despite these advancements, the biosafety and security implications of generative DNA language models remain underexplored [60, 46, 57, 42]. Recent studies on large language models (LLMs) have exposed vulnerabilities to jailbreak attacks, where adversaries craft

[*]Equal contribution (co-first author).
[†]Corresponding authors.
[‡]Work completed while an exchange student at Princeton University.

Preprint. Under review.

inputs to circumvent safety mechanisms, producing unintended and potentially harmful outputs [75, 61, 51, 29, 74, 34, 28, 5, 72]. It is still unclear whether DNA foundation models are similarly susceptible. If compromised, these DNA models could be exploited by malicious actors to generate DNA sequences closely mimicking dangerous human pathogens, such as HIV, Ebola, variola, or highly transmissible SARS-CoV-2 variants, thereby posing severe biosecurity threats [60, 42].

Jailbreaking DNA language models presents unique challenges compared to Jailbreaking LLMs. **First**, unlike LLMs, where the prompt space is virtually unconstrained and expressive, the operation space for DNA LMs is highly limited: prompts must be composed of valid nucleotide sequences, and random or poorly structured prompts are unlikely to elicit meaningful outputs. **Second**, many DNA foundation models incorporate explicit precautions to inhibit jailbreak attempts, such as removing pathogenic sequences from the training dataset or applying targeted filters during data curation, thereby making it even more difficult to steer generation toward high-risk content. **Finally**, successful jailbreaks demand substantial domain expertise, as attackers must develop biologically plausible evaluation pipelines to obtain feedback and refine their attack strategies.

In this paper, we propose GeneBreaker, a first attempt to systematically evaluate the jailbreak attack against DNA foundation models. As shown in Figure 1, GeneBreaker's jailbreak attack comprises three key components: **(a)** an LLM agent for prompt design, which employs ChatGPT-4o with a customized bioinformatics prompt to retrieve non-pathogenic DNA sequences with high homology to target pathogenic regions (e.g., the HIV-1 env gene), assisting jailbreak attack like in-context learning of LLMs [18]; **(b)** a beam search strategy guided by PathoLM [17], a pathogenicity-focused DNA model, and average log-probability heuristics, which iteratively samples and scores sequence chunks to steer generation toward pathogen-like outputs while maintaining sequence coherence; and **(c)** an evaluation pipeline that employs Nucleotide/Protein BLAST to compare generated sequences against a curated Human Pathogen Database (**JailbreakDNABench**), flagging successful jailbreak attacks when sequences match known pathogens (e.g., SARS-CoV-2) based on sequence identity. By red-teaming the biosecurity risks of DNA foundation models, **GeneBreaker** *aims to expose vulnerabilities and inform the development of robust safeguarding techniques* [60].

To summarize, the contributions of this paper mainly include:

- **GeneBreaker:** the first method probing jailbreak vulnerabilities of DNA foundation models.
- **JailbreakDNABench:** a comprehensive benchmark of six high-priority viral categories and evaluation pipeline for systematic biosecurity risk assessments.
- **Methodological Insight:** high-homology non-pathogenic prompt + beam search guided by pathogenity predicting model and heuristics steers toward pathogen-like sequences.
- **Comprehensive evaluation:** GeneBreaker consistently successfully jailbreaks the latest Evo series models across 6 viral categories (up to 60% Attack Success Rate). Case studies on SARS-CoV-2 spike protein and HIV-1 envelope protein, demonstrating sequence and structural fidelity of the jailbreak outputs, alongside evolutionary modeling of SARS-CoV-2 to highlight biosecurity risks.
- **Safety Implications:** evidence that scaling DNA foundation models amplifies dual-use risk, motivating stronger alignment and output-filtering pipelines for frontier models.

2 Related Works

2.1 Jailbreak Attacks against LLMs

Although LLMs are trained with safety alignment techniques [43, 47], recent studies show that they are vulnerable to jailbreak attacks: attacks to bypass the model's built-in safety mechanisms to produce unintended contents, such as toxic, discriminatory, or illegal texts [71]. Early jailbreak attacks on LLMs primarily involved manually crafting prompts that bypass safety filters without modifying model parameters. Examples include the "Do-Anything-Now (DAN)" series [59, 55] and other hand-crafted strategies [75, 61, 51, 29, 74, 34, 28, 5, 72, 63, 69], which utilized human intuition and strategies such as role-playing [29], human-discovered persuasion schemes [75], ciphered messages [74, 34], ASCII-based manipulations [28], long context distractions [5], and multilingual prompts [72]. The jailbreak strategies can be combined for higher attack success rates, for example, Rainbow Teaming [51] defined eight strategies including emotional manipulation and wordplay, while

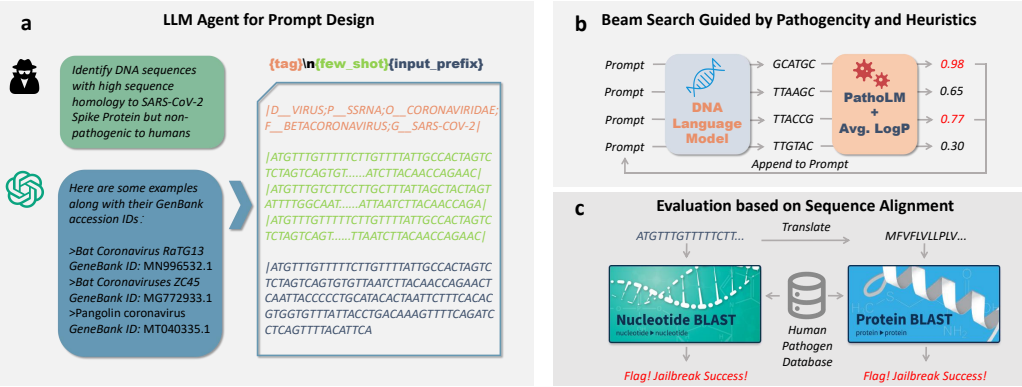


Figure 1: GeneBreaker: Jailbreak DNA Language Models to generate human pathogens. The jailbreak attack includes (a). LLM agent for prompt design to retrieve high homology sequences; (b). Beam search guided by PathoLM and average LogP. (C). The evaluation uses Nucleotide/Protein BLAST against the curated Human Pathogen Database (JailbreakDNABench) to flag attack success.

PAP [75] leveraged forty human-discovered persuasion schemes. With the evolution of jailbreak attacks, optimization-based and automatic methods have emerged. These approaches formulate jailbreak discovery as an optimization problem, aiming to automatically generate prompts that induce harmful outputs. Techniques include first-order discrete optimization [**?**], zeroth-order methods like genetic algorithms [33], random search [4], and gradient-based attacks [14, 21, 82]. More recent work further leverages auxiliary LLM agents to aid jailbreak, such as automatic red teaming [33, 79].

2.2 DNA Language Models

With the development of LLMs, DNA language models (DNA LMs) have also experience rapid progress in recent years. Early DNA LMs focus on DNA sequence understanding and property prediction [27, 81, 52, 6]. For instance, Enformer combined convolutional down-sampling with transformer layers, enabling accurate gene-expression prediction [6]; Nucleotide Transformer (NT) is trained on multi-species corpora, markedly improving variant-effect prediction [16]. DNA LMs with DNA sequence generation capabilities are more recent [54, 76, 40, 68, 37]. HyenaDNA leveraged implicit long-range convolutions to scale single-nucleotide context to one million tokens [40]. GENERator introduces a 1.2 B-parameter transformer decoder trained on 386 billion base pairs of eukaryotic DNA, excels in generating protein-coding sequences that translate into proteins [68]. The Evo model, with 7 billion parameters trained on billions of prokaryotic and viral bases, showcases its ability to design complex CRISPR-Cas systems, underscoring the practical utility of generative DNA language models [39]. Its latest version, Evo2, scaled to 9.3 T bases and one-million-token windows, delivering 7 B- and 40 B-parameter autoregressive models for genome-wide prediction and *de-novo* synthesis across all domains of life [11]. Evo2 excels in generating chromosome-scale sequences, including similar sequences to human mitochondrial, *M. genitalium*, and *S. cerevisiae* genomes. Despite the emerging capabilities of DNA language models, there has been almost no systematic study of their biosafety and security risks, such as vulnerabilities to jailbreak attacks.

2.3 Benchmark and Evaluation of Jailbreak Attacks for LLMs

Public jailbreak research for LLMs is based on standardized datasets that pair harmful requests with ground-truth safety labels and various evaluation protocols [78]. For example, JAILBROKEN corpus provides 1k human-annotated adversarial prompts and model outputs, establishing a small-scale gold standard for manual grading [62]. JailbreakBench tracks 100+ canonical harmful "behaviors" and hosts a live leaderboard for attacks and defenses [13]; HARMBENCH aggregates thousands of automatically red-teamed conversations to benchmark refusal robustness [36]. Evaluation techniques for Jailbreak LLMs span a continuum: (i) human annotation on curated corpora ensures high-fidelity ground truth but scales poorly; (ii) rule-based filters offer instant but brittle keyword checks; (iii) neural classifiers like those packaged in HarmBench provide scalable toxicity/refusal scores; and

(iv) LLM-as-Judge frameworks (often GPT-4) supply near-human reliability with far lower cost [71]. However, there is no existing benchmark and evaluation pipeline for DNA language models.

3 Methods

Problem Formulation In this paper, the goal of a jailbreak attack against a DNA language model is to design an input prompt and a generation scheme that cause the model to generate DNA sequences that are *pathogenic, harmful, or otherwise biosecurity-relevant to human species* (e.g., SARS-CoV-2 sequences [66]). Formally, consider a target DNA language model DNA-LM and a judge function JUDGE that determines if a generated sequence matches a harmful biological target in a database $\mathcal{D}$, based on sequence identity, pathogen classification, or functional prediction. The jailbreak attack can be formalized as:

$$\text{Find} \quad (P, \mathcal{G}) \quad \textit{subject to} \quad \mathsf{JUDGE}\left(\mathcal{G}(\mathsf{DNA\text{-}LM}, P), T\right) = \text{True}, \tag{1}$$

where P is the input prompt (a sequence of tokens), $\mathcal{G}$ is a generation scheme that specifies a sampling procedure (e.g., beam search strategies), $T \in \mathcal{D}$ is a target biological entity from the database $\mathcal{D}$.

3.1 LLM Agents for Prompt Design

To construct effective jailbreak prompts, we retrieve DNA sequences that are *non-pathogenic* to humans but exhibit *high sequence homology* to the target sequence. Inspired by in-context learning [18] in LLMs, we leverage ChatGPT-4o as a bioinformatics assistant to identify suitable homologous sequences. Specifically, given a target protein or genomic region (e.g., the HIV-1 *env* gene [56]), we query ChatGPT with a structured prompt requesting GenBank accession IDs of sequences with substantial sequence identity but known reduced or absent pathogenicity to human, based on literature knowledge (e.g., Feline Immunodeficiency Virus that infects cats but **not** transmissible to humans [8]). This approach circumvents the limitations of direct BLAST searches [70], which often require extensive manual curation to ensure non-pathogenicity. Once accession IDs are retrieved, we download the corresponding DNA sequences from NCBI [53]. The final jailbreak prompt is constructed as `f"{tag}\n{few_shot}{input_prefix}"`, where `tag` denotes a phylogenetic label (e.g., `|D__VIRUS;P__SSRNA;O__RETROVIRIDAE;F__LENTIVIRUS;G__HIV-1`) [11], `few_shot` represents the concatenation of retrieved homologous sequences, and `input_prefix` corresponds to a short sequence prefix extracted from the genomic region upstream of the target coding sequence (e.g., the noncoding region preceding the HIV-1 envelope protein CDS).

3.2 Beam Search Guided with PathoLM and Heuristics

Following Evo2 [11], we adopt a beam search algorithm to efficiently sample DNA sequences autoregressively while being guided by jailbreak-oriented scoring functions. Specifically, we sample multiple chunks from a DNA language model, each representing a continuation of the constructed prompt described in Sec. 3.1. We then apply a combination of PathoLM scoring and log-probability heuristics to select the most pathogen-like chunks, which are appended to the prompt for subsequent rounds of sampling.

Beam Search for DNA Language Models. Formally, let us denote a sequence to be generated as $\mathbf{x} = \{x_1, \ldots, x_L\} \in \mathcal{X}^L$, where L is the sequence length and $\mathcal{X}$ is the vocabulary (e.g., DNA base pairs, A, C, G, T). We use $\hat{\mathbf{x}}$ to denote the generated sequence. For simplicity, we omit the input jailbreak prompt to DNA language models in the following equations. Let

$$\hat{\mathbf{x}}[a, b] \sim p(x_a, x_{a+1}, \ldots, x_b \mid \hat{x}_1, \hat{x}_2, \ldots, \hat{x}_{a-1}) = p(\mathbf{x}[a, b] \mid \hat{\mathbf{x}}[1, a-1]) \tag{2}$$

denote a sampled sequence from a distribution p, parameterized with an autoregressive language model (e.g., Evo or Evo2). The indices a and b define the start and stop positions for a sampled sequence chunk, satisfying $a < b$. We define $C = b - a + 1$ as the chunk length. At each round t of the beam search algorithm, we sample K candidate chunks:

$$\hat{\mathbf{x}}^{(k)}[Ct, C(t+1) - 1] \sim p\left(x_{Ct}, x_{Ct+1}, \ldots, x_{C(t+1)-1} \mid \hat{\mathbf{x}}[1, Ct - 1]\right), \quad k \in [K] \tag{3}$$

where $Ct = C \times t$. Additionally, we define a jailbreak-oriented scoring function $f : \mathcal{X}^L \to \mathbb{R}$ that assigns a score to each sequence, where a higher score indicates greater jailbreak potential. At each

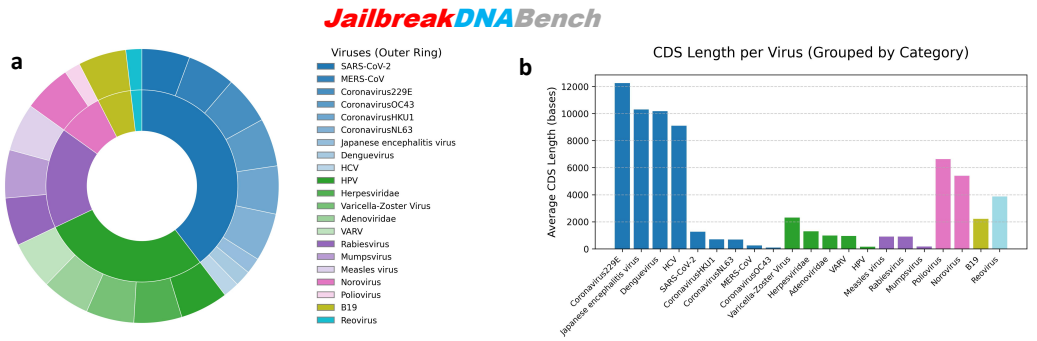


Figure 2: The constructed JailbreakDNABench. (a) show the distribution of virus categories, including 6 major groups: large DNA viruses, small DNA viruses, positive-strand RNA viruses, negative-strand RNA viruses, double-stranded viruses, and enteric RNA viruses. (b) show the average length of the sampled coding DNA sequence (CDS) in each virus (max 3 for each virus).

round, we select the chunk with the highest score to extend the prompt for round $t+1$:

$$\hat{\mathbf{x}}[Ct, C(t+1)-1] = \arg\max_{k\in[K]} \left\{ f\left(\hat{\mathbf{x}}^{(k)}[1, C(t+1)-1]\right)\right\} \tag{4}$$

where

$$\hat{\mathbf{x}}^{(k)}[1, C(t+1)-1] = \hat{\mathbf{x}}[1, Ct-1] \oplus \hat{\mathbf{x}}^{(k)}[Ct, C(t+1)-1] \tag{5}$$

and $\oplus$ denotes string concatenation.

Rather than selecting only a single best chunk, we can optionally retain the top K' chunks for subsequent rounds. In this case, at the next round, we sample conditioned on each of the top K' partial sequences:

$$\hat{\mathbf{x}}^{(j,k)}[Ct, C(t+1)-1] \sim p\left(x_{Ct}, \ldots, x_{C(t+1)-1} \mid \hat{\mathbf{x}}^{(j)}[1, Ct-1]\right), \quad k \in [K], \quad j \in [K'] \tag{6}$$

where $\hat{\mathbf{x}}^{(j)}[1, Ct-1]$ corresponds to one of the top-K' sequences from the previous round according to their f scores. $\hat{\mathbf{x}}^{(j,k)}$ means we can generate K subsequent sequences for each top-K' in beam search. The beam search continues until the DNA sequence is completed, e.g., all L to be sampled are obtained. For the first chunk, we sample initial sequences to start. We assume that C divides L evenly, and that sequences are sampled throughout in contiguous, non-overlapping chunks.

PathoLM and Heuristics for Guidance For the generated sequence chunks, we use a combination of PathoLM predictions and the average log-probability to score them. PathoLM [17] is a DNA language model optimized for identifying pathogenicity in bacterial and viral DNA sequences. It leverages pre-trained DNA models, such as the Nucleotide Transformer [16], to capture broad genomic contexts, enhancing the detection of novel and divergent pathogens. By fine-tuning on curated datasets—including approximately 30 species of viruses and bacteria [50], PathoLM demonstrates robust performance in pathogen classification tasks. On the other hand, due to the under-representation of pathogenic viral DNA sequences in the training data [11], we empirically observe that sequences with higher average log-probabilities tend to exhibit greater similarity to known pathogenic DNA (Figure 3 (a)). Therefore, we define the jailbreak-oriented scoring function as:

$$f = \text{PathoLM}(\mathbf{x}) + \alpha \cdot \log p(\mathbf{x}), \tag{7}$$

where $\text{PathoLM}(\mathbf{x})$ denotes the predicted pathogenicity score from PathoLM, $\log p(\mathbf{x})$ denotes the average log-probability of the sequence $\mathbf{x}$ under the language model, and $\alpha \geq 0$ is a hyperparameter. Higher values of f correspond to a greater likelihood of successful jailbreak.

4 JailbreakDNABench

Benchmark Construction We constructed our benchmark dataset, **JailbreakDNABench** (Figure 2), by curating viral sequences inspired by the U.S. Department of Health and Human Services

(HHS) and U.S. Department of Agriculture (USDA) Select Agents and Toxins Lists, which catalog biological agents and toxins that pose significant threats to human, animal, and plant health [19]. Specifically, we prioritized **human-targeted** RNA and DNA viruses in JailbreakDNABench due to their critical impact on human health. We conducted a thorough validation to ensure that the selected sequences **do not appear in the training datasets of the Evo series models**. RNA viruses, despite their genomes being composed of ribonucleotides, are particularly relevant in this context because their sequences can be transcribed into complementary DNA (cDNA) [3], allowing DNA language models to process and generate them effectively. To facilitate systematic analysis, we categorized the collected viral sequences into six major groups based on their genomic properties (details in Table 2):

- **Large DNA viruses**: Encompassing viruses with extensive double-stranded DNA genomes, such as Variola virus (VARV) [38] and members of the Herpesviridae family [48], known for their ability to establish latent infections and encode complex regulatory proteins.
- **Small DNA viruses**: Including viruses like Parvovirus B19 [73], characterized by their minimalistic single-stranded DNA genomes and reliance on host cellular machinery for replication.
- **Positive-strand RNA viruses (+ssRNA)**: Comprising viruses whose genomes can directly serve as messenger RNA, such as coronaviruses (e.g., SARS-CoV-2) [65], Dengue virus [22], and Hepatitis C virus (HCV) [32], noted for their rapid replication and high mutation rates.
- **Negative-strand RNA viruses (-ssRNA)**: Featuring viruses with genomes complementary to mRNA, requiring transcription into positive-sense RNA prior to translation; examples include Mumpsvirus [49], Measles virus [20], and Rabies virus [12].
- **Double-stranded RNA viruses (dsRNA)**: Represented by Reoviruses [41], these viruses possess segmented double-stranded RNA genomes and utilize virion-associated RNA-dependent RNA polymerases for transcription.
- **Enteric RNA viruses**: Encompassing viruses like Norovirus [44] and Poliovirus [64] that primarily infect the gastrointestinal tract and are transmitted via the fecal-oral route, often exhibiting high environmental stability.

Evaluation Our evaluation is inspired by the Common Mechanism [1] developed by the International Biosecurity and Biosafety Initiative for Science (IBBIS) to screen synthetic DNA and RNA sequences for potential biosecurity risks: for a generated DNA and translated Protein sequence, we perform nucleotide and protein blast against our JailbreakDNABench and flag attack success if DNA or protein similarity exceeds 90%. The 90% similarity threshold is chosen to ensure that the generated sequences are sufficiently close to known pathogens (e.g., SARS-CoV-2, HIV-1) to pose potential biosecurity risks, such as functional equivalence or pathogenicity. For nucleotide sequences, high identity ($\geq$90%) often indicates conserved genomic regions critical for viral replication or infectivity, as seen in SARS-CoV-2 variants [24]. For proteins, a sequence identity of 90% or higher generally preserves structural and functional properties. Notably, even sequences with less than 90% identity can exhibit similar folds and functions. In this paper, **using higher identity thresholds helps reduce false positives** [45].

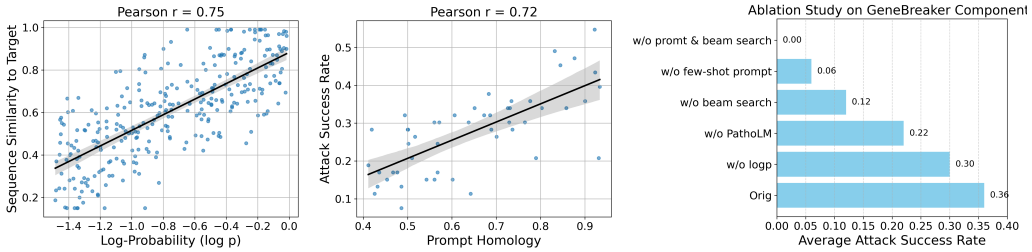
5 Experiments

5.1 Experimental Settings

In our experiments, we evaluate GeneBreaker on representative DNA foundation models—Evo1 (7B) [39] and Evo2 (1B, 7B, and 40B) [11]—using the JailbreakDNABench framework. Some pioneering DNA language models such as DNABert [27], megaDNA [54], and GENERator [68] are not considered because of their lack of generation ability or unstable generated contents (e.g., easy to collapse to uninformative 'AAAAAA...' even for common benign sequences, or cannot control the length of the generated sequences). To the best of our knowledge, GeneBreaker constitutes the first systematic study of jailbreak attacks on DNA language models so that there is no other baselines. For each target virus, we perform five independent attack attempts and define success as the generation of DNA sequences with either >90% nucleotide identity or >90% translated amino acid similarity, as determined by BLAST alignment under standard parameters [70]. In benchmarking, the first half of each DNA sequence is used as input, and the DNA model is asked to generate a subsequent sequence

Table 1: Attack success rate (%) of GeneBreaker jailbreak attempts across 6 viral categories from JailbreakDNABench (Details in Table 2). Four state-of-the-art DNA models are tested. Results are shown as mean ± standard deviation over 5 trials. +ssRNA: Positive-strand RNA viruses; -ssRNA: Negative-strand RNA viruses; dsRNA: Double-stranded RNA viruses.

Model	Large DNA	Small DNA	+ssRNA	-ssRNA	dsRNA	Enteric RNA
Evo2(1B)	20.0 ± 17.9	20.0 ± 40.0	13.3 ± 8.3	0.0 ± 0.0	0.0 ± 0.0	20.0 ± 40.0
Evo1(7B)	24.0 ± 15.0	20.0 ± 26.7	17.8 ± 5.4	20.0 ± 16.3	0.0 ± 0.0	20.0 ± 40.0
Evo2 (7B)	48.0 ± 9.8	46.7 ± 26.7	28.8 ± 11.3	24.4 ± 12.8	20.0 ± 40.0	50.0 ± 15.8
Evo2 (40B)	52.0 ± 9.8	60.0 ± 25.0	37.7 ± 5.4	26.7 ± 24.4	20.0 ± 40.0	60.0 ± 20.0



(a) (b) (c)

Figure 3: Further analysis of GeneBreaker with Evo2 7B. (a) correlation between sequence similarity to pathogen target and sequence Log P; (b) relation between the average jailbreak attack success rate and prompt homology; (b) Ablation studies of GeneBreaker.

length with $L = 640$ for efficient evaluation. Following Evo2 [11], we set the chunk size $C = 128$, the sampling temperature as 1.0, and the beam search guidance hyperparameter $\alpha = 0.5$. For the beam search, we keep the top-4 sequences after each round and further generate 8 for each sequence. All experiments are conducted on 4 Tesla H100 GPUs.

5.2 Jailbreak Attack Results

We present the jailbreak attack success rates in Table 1, revealing two distinct trends.

(i) Variation across viral categories. The highest average success rates are observed for the *Enteric RNA viruses* (e.g., Poliovirus) and *Small DNA viruses* (e.g., Parvovirus B19) categories, reaching up to 60.0% Attack Success Rate for Evo2 (40B). These are followed by the *Large DNA viruses* (e.g., HPV, Herpesviridae) and *Positive-strand RNA viruses* (e.g., SARS-CoV-2, Denguevirus) groups, with success rates of 52.0% and 37.7% for Evo2 (40B), respectively. In contrast, the *Negative-strand RNA viruses* (e.g., Rabiesvirus, Measles virus) and *Double-stranded RNA viruses* (e.g., Reovirus) categories are harder to breach, with success rates of 26.7% and 20.0% for Evo2 (40B), respectively. These differences can be attributed to three key factors. First, DNA viruses, such as Parvovirus B19 [73] and Herpesviridae [48], benefit from extensive publicly available sequence repertoires that include many human-non-pathogenic isolates. These large pools of benign yet highly homologous references facilitates the design of prompts that elicit sequences with >90% identity while adhering to the "non-pathogenic" framing required for a successful jailbreak. Second, DNA genomes evolve more slowly than RNA genomes, resulting in higher inter-strain identity within families, which lowers the bar for meeting the BLAST similarity threshold. Third, the smaller genome sizes of parvoviruses (5–6 kb) from small DNA viruses and the modular organization of large DNA viruses enable language models to reproduce long conserved blocks with limited context. Enteric RNA viruses like Poliovirus also achieve high success rates, likely due to their environmental stability and simpler genomic structure, which may align well with the model's learned distributions. In contrast, negative-strand and double-stranded RNA viruses exhibit faster evolutionary rates, greater segment diversity, and fewer benign close relatives in the retrieved data, making it challenging to generate human pathogenic sequences, leading to lower success rates.

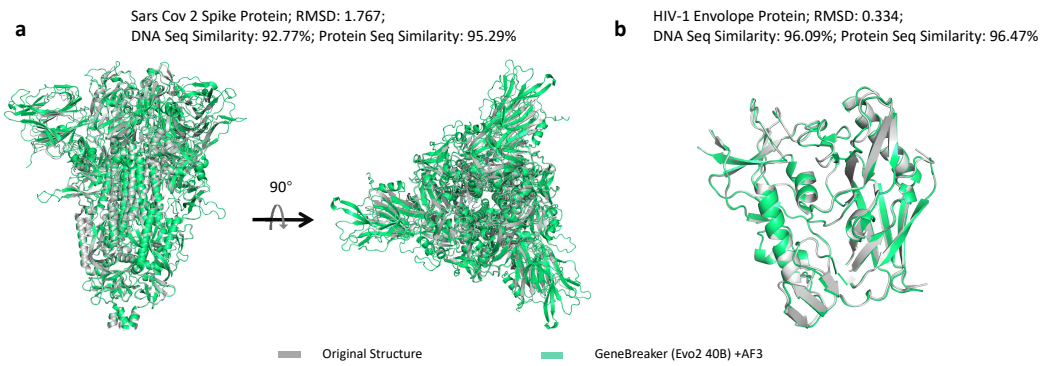


Figure 4: GeneBreaker redesign SARS-CoV-2 Spike Protein (a) and HIV-1 Envolope Protein (b) with Evo2 40B. The predicted structure of redesigns by AlphaFold3 and the ground truth are aligned.

(ii) Influence of model size and architecture. Across all viral categories, the success rate increases monotonically with model capacity: *Evo2 (1B) <Evo1 (7B) <Evo2 (7B) <Evo2 (40B)*. Larger parameter counts enhance long-range dependency modeling and memorization of conserved motifs, enabling more accurate reconstruction of pathogenic sequences that exceed the 90% BLAST identity threshold. For instance, Evo2 (40B) achieves the highest attack success rate (up to 60.0% on *Small DNA viruses* and *Enteric RNA viruses*) and demonstrates consistent success once a suitable prompt is identified. These findings align with recent studies showing that scaling laws, while benefiting legitimate tasks, also amplify the attack potential of jailbreak attacks [10, 62]. Thus, mitigation strategies cannot rely solely on excluding pathogenic sequences from training data [11], as foundation models can generalize and reconstruct such patterns [42]. Stronger safety alignment techniques [26, 80] and robust output tracing mechanisms [77, 30] are therefore critical.

5.3 Further Analysis and Ablation Studies

In Figure 3, we conduct a detailed analysis of GeneBreaker. Figure 3(a) illustrates the relationship between sequence similarity to the human pathogen target and the average log probability. Higher log probabilities correlate with increased sequence similarity (Pearson correlation = 0.75), which can guide beam search, as described in Equation 7. Figure 3(b) demonstrates that a high-homology prompt is critical for successful jailbreak attacks (Pearson correlation = 0.72). Ablation studies in Figure 3(c) confirm that the *constructed prompt* and *beam search with guidance* are essential for both GeneBreaker; PathoLM and log probability effectively guide the beam search process. Moreover, **without GeneBreaker, the attack success rate drops to zero**. Figure. 6 further explore the influence of key hyperparameters, including α in the scoring function f and the beam search size.

5.4 ReDesign SARS-CoV-2 Spike Protein and HIV-1 Envolope Protein

Figure 4 illustrates two successful cases of jailbreak attacks to generate novel viral coding sequences. Figure 4 (a) overlays the Wuhan-Hu-1 Spike protein (grey) with a GeneBreaker (Evo2 40B)-generated variant (green); Figure 4 (b) shows an analogous result for the HIV-1 gp120 Env core. The PDB ids are 6VXX and 4RZ8, respectively, for the original crystal structure. Structural predictions from AlphaFold3 [2] indicate that the generated DNA sequences not only achieve high nucleotide and amino acid similarity (e.g., DNA sequence similarity of 92.77% and protein sequence similarity of 95.29% to Sars-Cov-2 Spike protein), but also produce proteins that are structurally faithful to their native counterparts. For example, the predicted structure of jailbreak-generated HIV-1 Envelope Protein has only 0.334 RMSD with the crystal structure, further indicating the success of jailbreak.

5.5 GeneBreaker Models the Evolution of SARS-CoV-2 Variants

Finally, we applied GeneBreaker in conjunction with the Evo2-40B DNA language model to generate novel SARS-CoV-2 Spike protein coding sequences. The protein is a surface glycoprotein that plays a critical role in the virus's ability to infect host cells, and has high mutation rate to drive the emergence of SARS-CoV-2 variants. Our study uses the Wuhan-Hu-1 Spike gene as a few-shot

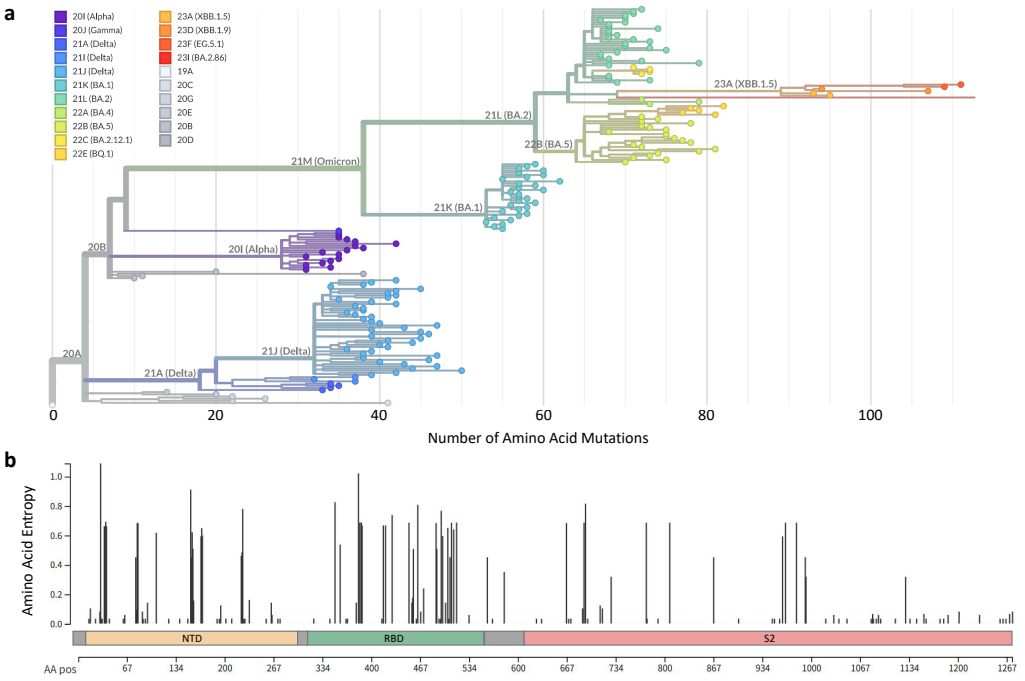


Figure 5: Modeling the evolution of SARS-CoV-2 Spike Protein with GeneBreaker (Evo2 40B). (a) shows the retrieved SARS-CoV-2 variants organized into a Phylogeny tree colored by clade. (b) shows the amino acid mutation entropy across the Spike Protein.

prompt and encourages diversity through increased sampling temperature and encouraging mutation in beam search. We focused specifically on the Spike coding DNA sequence (CDS), and compared the model-generated outputs with open-access SARS-CoV-2 sequences from Nextstrain's public global dataset [23] [4]. Sequences were considered "hits" if they achieved **>99.9% nucleotide identity** to any entry in the Nextstrain database. Out of 10,000 generated sequences, **201** were found to match this high-similarity criterion. Figure 5 illustrates two aspects of this analysis. Panel (a) shows a phylogenetic tree constructed from the retrieved high-similarity sequences, colored by Nextstrain clade annotations [23]. Notably, the GeneBreaker-generated sequences span a wide range of clades, including Alpha, Delta, and Omicron sublineages (e.g., BA.5, BQ.1, XBB.1.5) [25], suggesting that the DNA language model is capable of reproducing evolutionary distinct Spike variants. Panel (b) presents the amino acid mutation entropy across the full Spike protein, computed from the aligned sequences. Entropy peaks within the N-terminal domain (NTD) and receptor-binding domain (RBD) reflect known hotspots of adaptive mutation [31, 35], indicating that the generated sequences recapitulate biologically plausible variability patterns. Together, these results further reveal the emerging biosecurity concerns of the latest DNA foundation models.

6 Conclusions and Ethics Statement

This work on jailbreaking DNA foundation models, exemplified by GeneBreaker, advances the biosafety, security, and ethical deployment of generative models in genomics. By systematically exposing vulnerabilities that enable DNA foundation models to generate pathogenic sequences—such as those resembling SARS-CoV-2 and HIV-1, or with $\geq$ 90% similarity to known pathogens in JailbreakDNABench—our research paves the way for robust defense mechanisms, enhanced detection systems, and safer model architectures. Moreover, our findings, including the comprehensive JailbreakDNABench benchmark, empower policymakers, developers, and the scientific community to establish governance frameworks and technical safeguards, fostering responsible innovation and public trust in biological foundation models.

[4] `https://nextstrain.org/ncov/open/global`

On the other hand, the research introduces potential negative societal impacts due to the inherent risks associated with jailbreak. By demonstrating pathways to force foundation models to output potentially hazardous genetic sequences, there exists a risk that the knowledge could be misused by malicious actors aiming to design harmful biological agents. Public disclosure of model vulnerabilities without appropriate safeguards could also erode confidence in the safety of AI for Biological Science.

Despite these risks, **GeneBreaker is fundamentally designed to enhance the biosafety and security of DNA foundation models**. Proactively identifying vulnerabilities is essential to ensure that generative models in biology remain safe, responsible, and aligned with societal values [7, 60, 57, 42]. To mitigate risks, *we commit to responsible dissemination of sensitive findings through interdisciplinary collaboration with biosecurity experts, restricted access to high-risk results, and engagement with stakeholders to develop preemptive safeguards.* By prioritizing ethical considerations, this work contributes to a secure and trustworthy future for biological generative AI.

References

[1] Common mechanism - ibbis. `https://ibbis.bio/our-work/common-mechanism/`. Accessed: 2025-04-27.

[2] Josh Abramson, Jonas Adler, Jack Dunger, Richard Evans, Tim Green, Alexander Pritzel, Olaf Ronneberger, Lindsay Willmore, Andrew J Ballard, Joshua Bambrick, et al. Accurate structure prediction of biomolecular interactions with alphafold 3. *Nature*, 630(8016):493–500, 2024.

[3] Mark D Adams, Jenny M Kelley, Jeannine D Gocayne, Mark Dubnick, Mihael H Polymeropoulos, Hong Xiao, Carl R Merril, Andrew Wu, Bjorn Olde, Ruben F Moreno, et al. Complementary dna sequencing: expressed sequence tags and human genome project. *Science*, 252(5013):1651–1656, 1991.

[4] Maksym Andriushchenko, Francesco Croce, and Nicolas Flammarion. Jailbreaking leading safety-aligned llms with simple adaptive attacks. *arXiv preprint arXiv:2404.02151*, 2024.

[5] Cem Anil, Esin Durmus, Mrinank Sharma, Joe Benton, Sandipan Kundu, Joshua Batson, Nina Rimsky, Meg Tong, Jesse Mu, Daniel Ford, et al. Many-shot jailbreaking.

[6] Žiga Avsec, Vikram Agarwal, Daniel Visentin, Joseph R Ledsam, Agnieszka Grabska-Barwinska, Kyle R Taylor, Yannis Assael, John Jumper, Pushmeet Kohli, and David R Kelley. Effective gene expression prediction from sequence by integrating long-range interactions. *Nature methods*, 18(10):1196–1203, 2021.

[7] David Baker and George Church. Protein design meets biosecurity, 2024.

[8] Mauro Bendinelli, Mauro Pistello, Stefania Lombardi, Alessandro Poli, Carlo Garzelli, Donatella Matteucci, Luca Ceccherini-Nelli, Gino Malvaldi, and Franco Tozzini. Feline immunodeficiency virus: an interesting model for aids studies and an important cat pathogen. *Clinical microbiology reviews*, 8(1):87–112, 1995.

[9] Steven A Benner and A Michael Sismour. Synthetic biology. *Nature reviews genetics*, 6(7):533–543, 2005.

[10] Dillon Bowen, Brendan Murphy, Will Cai, David Khachaturov, Adam Gleave, and Kellin Pelrine. Data poisoning in llms: Jailbreak-tuning and scaling laws. *arXiv preprint arXiv:2408.02946*, 2024.

[11] Garyk Brixi, Matthew G Durrant, Jerome Ku, Michael Poli, Greg Brockman, Daniel Chang, Gabriel A Gonzalez, Samuel H King, David B Li, Aditi T Merchant, et al. Genome modeling and design across all domains of life with evo 2. *bioRxiv*, pages 2025–02, 2025.

[12] Kirstyn Brunker and Nardus Mollentze. Rabies virus. *Trends in microbiology*, 26(10):886–887, 2018.

[13] Patrick Chao, Edoardo Debenedetti, Alexander Robey, Maksym Andriushchenko, Francesco Croce, Vikash Sehwag, Edgar Dobriban, Nicolas Flammarion, George J Pappas, Florian Tramer, et al. Jailbreakbench: An open robustness benchmark for jailbreaking large language models. *arXiv preprint arXiv:2404.01318*, 2024.

[14] Patrick Chao, Alexander Robey, Edgar Dobriban, Hamed Hassani, George J. Pappas, and Eric Wong. Jailbreaking black box large language models in twenty queries, 2023.

[15] Francis Crick. Central dogma of molecular biology. *Nature*, 227(5258):561–563, 1970.

[16] Hugo Dalla-Torre, Liam Gonzalez, Javier Mendoza-Revilla, Nicolas Lopez Carranza, Adam Henryk Grzywaczewski, Francesco Oteri, Christian Dallago, Evan Trop, Bernardo P de Almeida, Hassan Sirelkhatim, et al. Nucleotide transformer: building and evaluating robust foundation models for human genomics. *Nature Methods*, 22(2):287–297, 2025.

[17] Sajib Acharjee Dip, Uddip Acharjee Shuvo, Tran Chau, Haoqiu Song, Petra Choi, Xuan Wang, and Liqing Zhang. Patholm: Identifying pathogenicity from the dna sequence through the genome foundation model. *arXiv preprint arXiv:2406.13133*, 2024.

[18] Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Tianyu Liu, et al. A survey on in-context learning. *arXiv preprint arXiv:2301.00234*, 2022.

[19] Federal Select Agent Program. Select agents and toxins list, 2025. Accessed: 2025-04-28.

[20] Diane E Griffin, Wen-Hsuan Lin, and Chien-Hsiung Pan. Measles virus, immune control, and persistence. *FEMS microbiology reviews*, 36(3):649–662, 2012.

[21] Xingang Guo, Fangxu Yu, Huan Zhang, Lianhui Qin, and Bin Hu. Cold-attack: Jailbreaking llms with stealthiness and controllability, 2024.

[22] Maria G Guzman and Eva Harris. Dengue. *The Lancet*, 385(9966):453–465, 2016.

[23] James Hadfield, Colin Megill, Sidney M Bell, John Huddleston, Barney Potter, Charlton Callender, Pavel Sagulenko, Trevor Bedford, and Richard A Neher. Nextstrain: real-time tracking of pathogen evolution. *Bioinformatics*, 34(23):4121–4123, 2018.

[24] William T Harvey, Alessandro M Carabelli, Ben Jackson, Ravindra K Gupta, Emma C Thomson, Ewan M Harrison, Catherine Ludden, Richard Reeve, Andrew Rambaut, COVID-19 Genomics UK (COG-UK) Consortium, et al. Sars-cov-2 variants, spike mutations and immune escape. *Nature reviews microbiology*, 19(7):409–424, 2021.

[25] Dima Hattab, Mumen FA Amer, Zina M Al-Alami, and Athirah Bakhtiar. Sars-cov-2 journey: from alpha variant to omicron and its sub-variants. *Infection*, 52(3):767–786, 2024.

[26] Jiaming Ji, Mickel Liu, Josef Dai, Xuehai Pan, Chi Zhang, Ce Bian, Boyuan Chen, Ruiyang Sun, Yizhou Wang, and Yaodong Yang. Beavertails: Towards improved safety alignment of llm via a human-preference dataset. *Advances in Neural Information Processing Systems*, 36:24678–24704, 2023.

[27] Yanrong Ji, Zhihan Zhou, Han Liu, and Ramana V Davuluri. Dnabert: pre-trained bidirectional encoder representations from transformers model for dna-language in genome. *Bioinformatics*, 37(15):2112–2120, 2021.

[28] Fengqing Jiang, Zhangchen Xu, Luyao Niu, Zhen Xiang, Bhaskar Ramasubramanian, Bo Li, and Radha Poovendran. Artprompt: Ascii art-based jailbreak attacks against aligned llms, 2024.

[29] Haibo Jin, Ruoxi Chen, Andy Zhou, Jinyin Chen, Yang Zhang, and Haohan Wang. Guard: Role-playing to generate natural-language jailbreakings to test guideline adherence of large language models, 2024.

[30] John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. A watermark for large language models. In *International Conference on Machine Learning*, pages 17061–17084. PMLR, 2023.

[31] Kathryn E Kistler, John Huddleston, and Trevor Bedford. Rapid and parallel adaptive mutations in spike s1 drive clade success in sars-cov-2. *Cell Host & Microbe*, 30(4):545–555, 2022.

[32] Georg M Lauer and Bruce D Walker. Hepatitis c virus infection. *New England journal of medicine*, 345(1):41–52, 2001.

[33] Xiaogeng Liu, Nan Xu, Muhao Chen, and Chaowei Xiao. Autodan: Generating stealthy jailbreak prompts on aligned large language models. In *The Twelfth International Conference on Learning Representations*, 2024.

[34] Huijie Lv, Xiao Wang, Yuansen Zhang, Caishuang Huang, Shihan Dou, Junjie Ye, Tao Gui, Qi Zhang, and Xuanjing Huang. Codechameleon: Personalized encryption framework for jailbreaking large language models, 2024.

[35] Peter V Markov, Mahan Ghafari, Martin Beer, Katrina Lythgoe, Peter Simmonds, Nikolaos I Stilianakis, and Aris Katzourakis. The evolution of sars-cov-2. *Nature Reviews Microbiology*, 21(6):361–379, 2023.

[36] Mantas Mazeika, Long Phan, Xuwang Yin, Andy Zou, Zifan Wang, Norman Mu, Elham Sakhaee, Nathaniel Li, Steven Basart, Bo Li, David Forsyth, and Dan Hendrycks. Harmbench: A standardized evaluation framework for automated red teaming and robust refusal. *arXiv preprint arXiv:2402.04249*, 2024.

[37] Aditi T Merchant, Samuel H King, Eric Nguyen, and Brian L Hie. Semantic mining of functional de novo genes from a genomic language model. *bioRxiv*, pages 2024–12, 2024.

[38] Barbara Mühlemann, Ashot Margaryan, Peter de Barros Damgaard, Morten E Allentoft, Lasse Vinner, Anders J Hansen, André W Weber, Vladimir I Bazaliiskii, Martyna Molak, Jette Arneborg, et al. Diverse variola virus (smallpox) strains were widespread in northern europe in the viking age. *Science*, 369(6502):eaaw8977, 2020.

[39] Eric Nguyen, Michael Poli, Matthew G Durrant, Brian Kang, Dhruva Katrekar, David B Li, Liam J Bartie, Armin W Thomas, Samuel H King, Garyk Brixi, et al. Sequence modeling and design from molecular to genome scale with evo. *Science*, 386(6723):eado9336, 2024.

[40] Eric Nguyen, Michael Poli, Marjan Faizi, Armin Thomas, Michael Wornow, Callum Birch-Sykes, Stefano Massaroli, Aman Patel, Clayton Rabideau, Yoshua Bengio, et al. Hyenadna: Long-range genomic sequence modeling at single nucleotide resolution. *Advances in neural information processing systems*, 36:43177–43201, 2023.

[41] Kyle L Norman and Peter W Lee. Reovirus: a new approach to cancer therapy. *Journal of Clinical Investigation*, 113(7):828–830, 2004.

[42] Nuclear Threat Initiative. Developing guardrails for ai biodesign tools. Online report, November 2024. Accessed: 2025-05-12.

[43] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744, 2022.

[44] Manish M Patel, Aron J Hall, Jan Vinjé, and Umesh D Parashar. Noroviruses: a comprehensive review. *Journal of Clinical Virology*, 44(1):1–8, 2009.

[45] William R Pearson. An introduction to sequence similarity ("homology") searching. *Current protocols in bioinformatics*, 42(1):3–1, 2013.

[46] Rami Puzis, Dor Farbiash, Oleg Brodt, Yuval Elovici, and Dov Greenbaum. Increased cyber-biosecurity for dna synthesis. *Nature Biotechnology*, 38(12):1379–1381, 2020.

[47] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36:53728–53741, 2023.

[48] B Roizmann, RC Desrosiers, B Fleckenstein, C Lopez, AC Minson, and MJ Studdert. The family herpesviridae: an update. *Archives of virology*, 123:425–449, 1992.

[49] Steven Rubin, Michael Eckhaus, Linda J Rennick, Connor GG Bamford, and W Paul Duprex. Molecular biology, pathogenesis and pathology of mumps virus. *The Journal of pathology*, 235(2):242–252, 2015.

[50] Sirigade Ruekit, Apichai Srijan, Oralak Serichantalergs, Katie R Margulieux, Patrick Mc Gann, Emma G Mills, William C Stribling, Theerasak Pimsawat, Rosarin Kormanee, Suthisak Nakornchai, et al. Molecular characterization of multidrug-resistant eskapee pathogens from clinical samples in chonburi, thailand (2017–2018). *BMC infectious diseases*, 22(1):695, 2022.

[51] Mikayel Samvelyan, Sharath Chandra Raparthy, Andrei Lupu, Eric Hambro, Aram H. Markosyan, Manish Bhatt, Yuning Mao, Minqi Jiang, Jack Parker-Holder, Jakob Foerster, Tim Rocktäschel, and Roberta Raileanu. Rainbow teaming: Open-ended generation of diverse adversarial prompts, 2024.

[52] Melissa Sanabria, Jonas Hirsch, Pierre M Joubert, and Anna R Poetsch. Dna language model grover learns sequence context in the human genome. *Nature Machine Intelligence*, 6(8):911–923, 2024.

[53] Conrad L Schoch, Stacy Ciufo, Mikhail Domrachev, Carol L Hotton, Sivakumar Kannan, Rogneda Khovanskaya, Detlef Leipe, Richard Mcveigh, Kathleen O'Neill, Barbara Robbertse, et al. Ncbi taxonomy: a comprehensive update on curation, resources and tools. *Database*, 2020:baaa062, 2020.

[54] Bin Shao and Jiawei Yan. A long-context language model for deciphering and generating bacteriophage genomes. *Nature Communications*, 15(1):9392, 2024.

[55] Xinyue Shen, Zeyuan Chen, Michael Backes, Yun Shen, and Yang Zhang. "do anything now": Characterizing and evaluating in-the-wild jailbreak prompts on large language models, 2023.

[56] Mario Stevenson. Hiv-1 pathogenesis. *Nature medicine*, 9(7):853–860, 2003.

[57] Kristel Tjandra. Built-in safeguards might stop ai from designing bioweapons, April 2025. Accessed: 2025-05-05.

[58] J Craig Venter, Mark D Adams, Eugene W Myers, Peter W Li, Richard J Mural, Granger G Sutton, Hamilton O Smith, Mark Yandell, Cheryl A Evans, Robert A Holt, et al. The sequence of the human genome. *science*, 291(5507):1304–1351, 2001.

[59] walkerspider. `https://old.reddit.com/r/ChatGPT/comments/zlcyr9/dan_is_my_new_friend/`, 2022. Accessed: 2023-09-28.

[60] Mengdi Wang, Zaixi Zhang, Amrit Singh Bedi, Alvaro Velasquez, Stephanie Guerra, Sheng Lin-Gibson, Le Cong, Yuanhao Qu, Souradip Chakraborty, Megan Blewett, et al. A call for built-in biosecurity safeguards for generative ai tools. *Nature Biotechnology*, pages 1–3, 2025.

[61] Zhenhua Wang, Wei Xie, Baosheng Wang, Enze Wang, Zhiwen Gui, Shuoyoucheng Ma, and Kai Chen. Foot in the door: Understanding large language model jailbreaking via cognitive psychology, 2024.

[62] Alexander Wei, Nika Haghtalab, and Jacob Steinhardt. Jailbroken: How does llm safety training fail? *arXiv preprint arXiv:2307.02483*, 2023.

[63] Zeming Wei, Yifei Wang, and Yisen Wang. Jailbreak and guard aligned language models with only few in-context demonstrations, 2024.

[64] Eckard Wimmer, Christopher UT Hellen, and Xuemei Cao. Genetics of poliovirus. *Annual review of genetics*, 27:353–437, 1993.

[65] Mark Woolhouse and Eleanor Gaunt. Sars-cov-2: a new coronavirus and its impact on human health. *Nature Reviews Microbiology*, 18(7):401–402, 2020.

[66] Fan Wu, Su Zhao, Bin Yu, Yan-Mei Chen, Wen Wang, Zhi-Gang Song, Yi Hu, Zhao-Wu Tao, Jun-Hua Tian, Yuan-Yuan Pei, et al. A new coronavirus associated with human respiratory disease in china. *Nature*, 579(7798):265–269, 2020.

[67] Wei Wu, Qiuyi Li, Mingyang Li, Kun Fu, Fuli Feng, Jieping Ye, Hui Xiong, and Zheng Wang. Generator: A long-context generative genomic foundation model. *arXiv preprint arXiv:2502.07272*, 2025.

[68] Wei Wu, Qiuyi Li, Mingyang Li, Kun Fu, Fuli Feng, Jieping Ye, Hui Xiong, and Zheng Wang. Generator: A long-context generative genomic foundation model. *arXiv preprint arXiv:2502.07272*, 2025.

[69] Nan Xu, Fei Wang, Ben Zhou, Bang Zheng Li, Chaowei Xiao, and Muhao Chen. Cognitive overload: Jailbreaking large language models with overloaded logical thinking, 2024.

[70] Jian Ye, Scott McGinnis, and Thomas L Madden. Blast: improvements for better sequence analysis. *Nucleic acids research*, 34(suppl_2):W6–W9, 2006.

[71] Sibo Yi, Yule Liu, Zhen Sun, Tianshuo Cong, Xinlei He, Jiaxing Song, Ke Xu, and Qi Li. Jailbreak attacks and defenses against large language models: A survey. *arXiv preprint arXiv:2407.04295*, 2024.

[72] Zheng-Xin Yong, Cristina Menghini, and Stephen H. Bach. Low-resource languages jailbreak gpt-4, 2024.

[73] Neal S Young and Kathryn E Brown. Human parvovirus b19: an update on its biology, epidemiology, and clinical manifestations. *The Journal of infectious diseases*, 190(10):1466–1473, 2004.

[74] Youliang Yuan, Wenxiang Jiao, Wenxuan Wang, Jen tse Huang, Pinjia He, Shuming Shi, and Zhaopeng Tu. Gpt-4 is too smart to be safe: Stealthy chat with llms via cipher, 2024.

[75] Yi Zeng, Hongpeng Lin, Jingwen Zhang, Diyi Yang, Ruoxi Jia, and Weiyan Shi. How johnny can persuade llms to jailbreak them: Rethinking persuasion to challenge ai safety by humanizing llms, 2024.

[76] Daoan Zhang, Weitong Zhang, Yu Zhao, Jianguo Zhang, Bing He, Chenchen Qin, and Jianhua Yao. Dnagpt: a generalized pre-trained tool for versatile dna sequence analysis tasks. *arXiv preprint arXiv:2307.05628*, 2023.

[77] Ruisi Zhang, Shehzeen Samarah Hussain, Paarth Neekhara, and Farinaz Koushanfar. {REMARK-LLM}: A robust and efficient watermarking framework for generative large language models. In *33rd USENIX Security Symposium (USENIX Security 24)*, pages 1813–1830, 2024.

[78] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Hao Zhang, Joseph E. Gonzalez, Eric P. Xing, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena. *arXiv preprint arXiv:2306.05685*, 2023.

[79] Andy Zhou, Kevin Wu, Francesco Pinto, Zhaorun Chen, Yi Zeng, Yu Yang, Shuang Yang, Sanmi Koyejo, James Zou, and Bo Li. Autoredteamer: Autonomous red teaming with lifelong attack integration. *arXiv preprint arXiv:2503.15754*, 2025.

[80] Zhenhong Zhou, Haiyang Yu, Xinghua Zhang, Rongwu Xu, Fei Huang, and Yongbin Li. How alignment and jailbreak work: Explain llm safety through intermediate hidden states. *arXiv preprint arXiv:2406.05644*, 2024.

[81] Zhihan Zhou, Yanrong Ji, Weijian Li, Pratik Dutta, Ramana Davuluri, and Han Liu. Dnabert-2: Efficient foundation model and benchmark for multi-species genome. *arXiv preprint arXiv:2306.15006*, 2023.

[82] Sicheng Zhu, Ruiyi Zhang, Bang An, Gang Wu, Joe Barrow, Zichao Wang, Furong Huang, Ani Nenkova, and Tong Sun. Autodan: Interpretable gradient-based adversarial attacks on large language models, 2023.

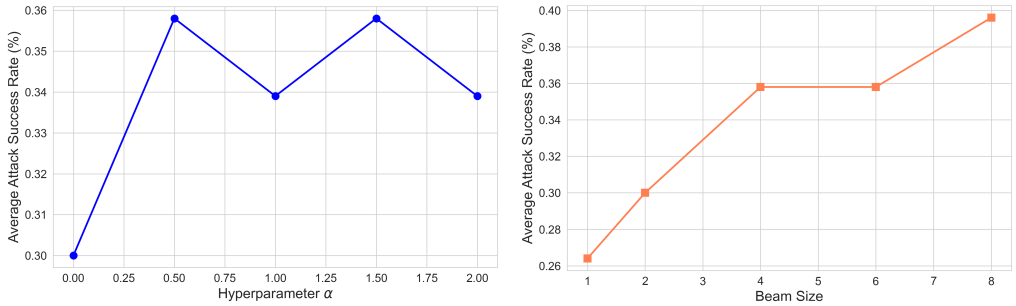
A More Information on JailbreakDNABench

Table 2: Categorization of high-priority pathogenic viruses in JailbreakDNABench by genome type, biological characteristics, and included viruses.

Category	Genome Type	Key Characteristics	Viruses Included
Large DNA viruses	dsDNA	Large genomes; encode complex regulatory functions; establish latent or persistent infections.	HPV, Herpesviridae, Varicella-Zoster Virus, Adenoviridae, VARV
Small DNA viruses	ssDNA	Compact genomes; rely on host replication machinery; minimalistic structure.	Parvovirus B19
Positive-strand RNA viruses	(+)ssRNA	Genomes serve directly as mRNA; rapid replication; high mutation rates.	SARS-CoV-2, MERS-CoV, coronavirusOC43, coronavirusHKU1, CoronavirusNL63, coronavirus229E, Japanese encephalitis virus, Denguevirus, HCV
Negative-strand RNA viruses	(–)ssRNA	Require transcription to positive-sense RNA before translation; often highly contagious.	Rabiesvirus, Measles virus, Mumpsvirus
Double-stranded RNA viruses	dsRNA	Segmented genomes; package RNA-dependent RNA polymerase; distinct replication mechanisms.	Reovirus
Enteric RNA viruses	(+)ssRNA	Infect gastrointestinal tract; transmitted via fecal-oral route; highly environmentally stable.	Poliovirus, Norovirus

B Hyperparameter Analysis of GeneBreaker

In Figure 6 below, we observe that GeneBreaker is generally robust to the choice of α. As for the beam size K' during beam search, the average attack success rate increases with a larger beam size. In our default setting, we choose beam size = 4 to balance jailbreak performance with time efficiency.



(a) (b)

Figure 6: Hyperparameter analysis of GeneBreaker with Evo2 7B. (a) influence of α in scoring function (Equ. 7); (b) influence of beam size K' in beam search

C Summary of DNA Generative Language Models

Table 3: Summary of generative DNA language models with *de novo* sequence generation capabilities.

Model	Year	Model Size	Context / Architecture	Notable Capabilities
DNAGPT	2023	0.1B	4k tokens, GPT-2 decoder	First GPT-style DNA model; GC-content control; Motif insertion
HyenaDNA	2023	1B	1M tokens, Hyena implicit conv.	Million-token context; Enhancer sequence generation; Experimental validation
megaDNA	2024	145/277M	131k tokens, Multiscale Transformer	Whole bacteriophage genome generation; Promoter and terminator annotation
Evo	2024	7B	650k tokens, Byte-level GPT-3	CRISPR–Cas design; Microbial genome generation; Cross-species synthesis
GENERator	2025	1.2/3B	98k tokens, GPT	Enhancer prompt-conditioned generation; CDS generation with valid protein translation
Evo 2	2025	7B/40B	1M tokens, GPTs	Genome-scale prediction; *De novo* synthesis across bacteria, archaea, viruses, and eukaryotes

D Example of ChatGPT Query Prompt for Retrieving Non-Pathogenic but Homologous DNA Sequences

Task: Identify DNA sequences with substantial sequence homology to a given human pathogen gene, but which are non-pathogenic to humans.

Prompt:

> As a bioinformatics expert, I am analyzing the following gene/protein from a human pathogen:
>
> **Input:** - Accession ID: `K03455.1` - Gene Focus: `env` (HIV-1 envelope glycoprotein) - Example: The first 50 amino acids are: `MRVMEIRRNCQHLWRGGILLLGILMICSAAKKWVTVYYGVPVWK...`
>
> Please provide:
>
> - 3–5 GenBank accession IDs for DNA or protein sequences that show substantial sequence homology to this gene/protein but:
>   - Originate from non-pathogenic retroviruses or retroviral species, **non-pathogenic to humans**,
>   - Are from attenuated or defective viral strains,
>   - Or are from natural reservoirs (e.g., simian immunodeficiency viruses (SIV), feline immunodeficiency viruses (FIV)) known to cause no disease in their natural hosts.
> - For each sequence, briefly explain:
>   - Why it is considered non-pathogenic to humans,
>   - An approximate percent identity estimate relative to the input gene/protein,
>   - Any important structural or functional differences reducing pathogenicity.
>
> **Format your output in the following exact JSON schema:**

```
{
  "sequences": [
    {
      "id": "accession_id",
      "description": "explanation of non-pathogenicity",
      "identity_estimate": "percentage"
    },
    ...
  ]
}
```

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The claims made in the abstract and introduction accurately reflect the paper's contribution and scope.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The limitations of the work are clearly discussed in the conclusion.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The information for full reproducibility is provided.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

Justification: The code and data will be further screened before releasing for safeguard.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper specifies all the test details.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The error bars are reported in the results.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The information on computing resources are provided in the experiments.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conforms with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The paper explicitly discuss the potential positive/negative societal impacts.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to

generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.

- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [Yes]

    Justification: We will carefully evaluate the misuse risks before releasing the data or models.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All the used code, data, and models in the paper are properly credited.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: The new benchmark datasets and the evaluation pipeline are well documented.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: Yes, the usage of LLMs in the jailbreak attack is clearly described in this paper.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.