GenoArmory: A Unified Evaluation Framework for Adversarial Attacks on Genomic Foundation Models

Anonymous Author(s)
Affiliation
Address
email

Abstract

We propose the **first** unified adversarial attack benchmark for Genomic Foundation Models (GFMs), named **GenoArmory**. Unlike existing GFM benchmarks, GenoArmory offers the first comprehensive evaluation framework to systematically assess the vulnerability of GFMs to adversarial attacks. Methodologically, we evaluate the adversarial robustness of five state-of-the-art GFMs using four widely adopted attack algorithms and three defense strategies. Importantly, our benchmark provides an accessible and comprehensive framework to analyze GFM vulnerabilities with respect to model architecture, quantization schemes, and training datasets. Additionally, we introduce **GenoAdv**, a new adversarial sample dataset designed to improve GFM safety. Empirically, classification models exhibit greater robustness to adversarial perturbations compared to generative models, highlighting the impact of task type on model vulnerability. Moreover, adversarial attacks frequently target biologically significant genomic regions, suggesting that these models effectively capture meaningful sequence features.

1 Introduction

The advent of Genomic Foundation Models (GFMs) has revolutionized the analysis and generation of DNA and RNA sequences [96, 95, 94, 85, 63, 12, 64, 30]. These models, pre-trained on extensive genomic datasets, have demonstrated exceptional performance across a variety of genomics tasks, leading to widespread adoption in both research and industry. For instance, GFMs have shown proficiency in generating high-quality DNA and RNA sequences [96, 63] and in species classification tasks [94, 12, 30]. In the realm of medical diagnostics, GFMs contribute significantly by predicting gene pathogenicity [70] and assessing genome-wide variant effects [3]. Their capabilities extend to functional genomics, aiding in promoter detection [21] and transcription factor prediction [23, 35], which are crucial for understanding gene regulation mechanisms. GFMs also are instrumental in RNA secondary structure prediction [82], a critical aspect of understanding RNA function and interactions.

Despite the remarkable advancements, GFMs face significant challenges, particularly concerning their robustness and security. GFMs, which process structured, high-dimensional, and low-redundancy inputs like DNA sequences, are especially susceptible to adversarial attacks—even minor perturbations, such as single-nucleotide variations, can lead to substantial biological consequences. For instance, recent studies [58] have demonstrated that DNA language models, including DNABERT-2 and the Nucleotide Transformer, are vulnerable to various adversarial strategies including nucleotide-level substitutions, codon-level modifications, and backtranslation-based transformations. Such attacks can significantly degrade model performance in tasks like antimicrobial resistance gene classification and promoter detection. Moreover, the generative capabilities of GFMs can be exploited by the attacker—it could manipulate models like GenomeOcean [96] to produce biologically nonsensical sequences, potentially leading to harmful application, even including the design of bioweapons [67].

Given the significant safety concerns surrounding GFMs, there is a pressing need for robust defense mechanisms to ensure their reliability and security. However, the absence of benchmarks specifically designed to evaluate GFM safety has hindered the development of effective defense methods. Existing

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

Figure 1: **An overview of benchmarking adversarial attacks on GFMs**

efforts [94, 52] primarily assess performance, without addressing safety aspects. This highlights the urgency of developing a new benchmark specifically designed to evaluate the safety of GFMs. To address this need, we introduce the GenoArmory benchmark, as shown in Figure 1, designed to standardize best practices in the emerging field of adversarial attack and defense for DNA-based GFMs. GenoArmory is guided by core principles of transparency, reproducibility, and fairness in evaluating GFM robustness under both attack and defense scenarios. In this paper, we detail these guiding principles, describe the benchmark's components, report results across multiple attack and defense strategies on various GFMs, and share insights to inform robustness improvements.

Contributions: We propose the GenoArmory framework (Figure 2) to a comprehensively assess the robustness of GFMs against adversarial attacks. Our contributions include:

- **Pipeline for red-teaming GFMs.** We present a comprehensive evaluation pipeline to assess the robustness of DNA-based GFMs against adversarial attacks. Specifically, our pipeline implements both gradient-based and gradient-free attack strategies across five different GFMs with standardized evaluation metrics.
- **Pipeline for testing and adding new defenses.** We implement three defense mechanisms and evaluate their effectiveness against adversarial attacks. Additionally, we provide plug-and-play code to enable standardized evaluation of newly developed defense methods.
- **Repository of GFM adversarial attack artifacts.** We provide a repository of adversarial attack artifacts on GFMs, including adversarial examples and attack code, to facilitate reproducibility and further research in this area.
- **New adversarial sample dataset for GFMs.** We introduce a new dataset **GenoAdv**, composed of adversarial examples specifically generated to improve the robustness of GFMs. When used in training, GenoAdv yield a **34.71%** Defense Success Rate, compared to training using only TextFooler samples.
- **Meaningful insights.** We provide a comprehensive analysis of GFM robustness under adversarial attacks, revealing the strengths and limitations of various models and defense strategies. Additionally, we offer an in-depth discussion on how training methods and quantization settings impact the robustness of GFMs.

2 Background

Definition. Given a genomic sequence $X = [x_1, x_2, \ldots, x_n]$, where each nucleotide $x_i \in \{A, T, C, G\}$, a DNA model $f(\cdot)$, and a corresponding label y, our goal is to find an adversarial sequence X' that satisfies:

$$f(X') \neq y \quad \text{subject to} \quad d(X, X') \leq \epsilon,$$

where $d(\cdot, \cdot)$ is a distance metric measuring the perturbation between the original and adversarial sequences, and ϵ controls the perturbation budget.

Genomic Foundation Models. Recent advances in genomic foundation models (GFMs) [52] establish two principal methodological paradigms: classification models and generative models. Within the classification paradigm, transformer-based approaches exhibit progressive technical refinements. Initial models, including DNABERT [30] and Nucleotide Transformer [12], establish baseline performance through fixed k-mer tokenization strategies. DNABERT-2 [94] addresses these constraints by integrating byte-pair encoding (BPE) for tokenization and Attention with Linear Biases

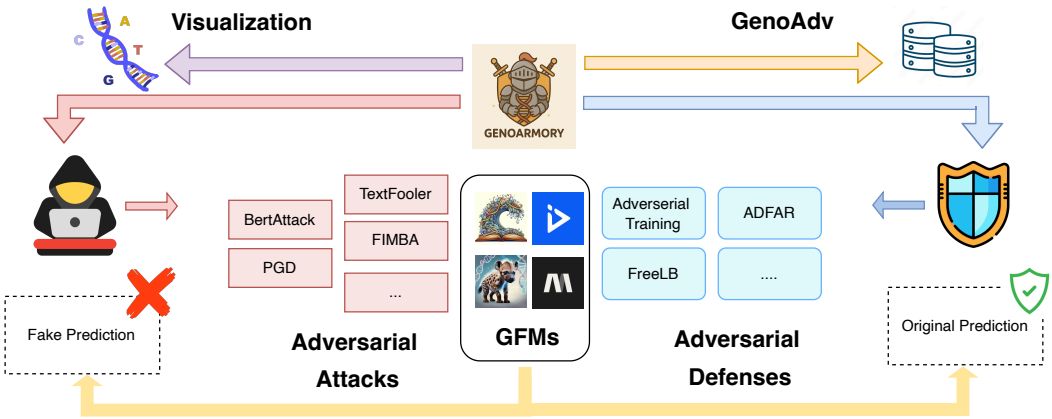

Figure 2: **GenoArmory Framework.** Our GenoArmory framework incorporates diverse adversarial attack and defense methods on GFMs. It also offers visualization tools to highlight important regions influencing model predictions and introduces a new adversarial dataset, **GenoAdv**.

(ALiBi) for modeling longer sequences, which significantly enhances motif discovery capabilities. Building on this, DNABERT-S [95] focuses on species differences in the embedding space. GERM [55] emerges as the first GFM specifically optimized for resource-constrained environments. By integrating an outlier-free architecture, GERM achieves both reliable quantization and fast adaptation. For long-range genomic dependency modeling, HyenaDNA [64] replaces conventional attention mechanisms with Hyena operators, enabling efficient processing of ultra-long genomic sequences. Among generative models, GenomeOcean [96] represents a pioneer, trains on 220TB of genomic data, and demonstrates strong DNA sequence generation capabilities across diverse species domains. Meanwhile, Evo [63] introduces a hybrid architecture that combines Hyena operators with sparse attention mechanisms capable of performing whole-genome modeling at single nucleotide resolution.

Attack Methods. As shown in Figure 5, adversarial attacks are broadly categorized into untargeted, targeted, and universal variants. Untargeted attacks [51, 56] aim to maximize model loss by perturbing inputs toward the gradient, while targeted attacks [5, 90] steer predictions toward specific classes by gradient. Universal attacks [60] generate input-agnostic perturbations that mislead models across entire data distributions. Numerous adversarial attack methods have been proposed in both NLP and CV, demonstrating their effectiveness in impacting model performance. Only one work, FIMBA [74], propose adversarial attacks in the genomic domain. FIMBA introduces a black-box, model-agnostic framework that perturbs key features identified via SHAP values to disrupt genomic models.

Defense Methods. As shown in Figure 5, defense strategies are broadly categorized into adversarial training, defensive distillation, adversarial sample detection, and regularization with certified robustness. Adversarial training [97, 56] enhances model robustness by iteratively injecting adversarial examples during training, Another approach defensive distillation [66] trains student models on softened probability distributions from teacher models to smooth decision boundaries. In contrast, adversarial sample [34, 93, 69] detection identifies malicious inputs at inference time. Regularization with certified robustness [43, 50, 84, 31] reduces vulnerability through loss shaping.

3 Main Features for GenoArmory

Given the current landscape of GFMs, there exists no benchmark dedicated to evaluating their reliability. Considering the significant safety concerns, we propose the **first** benchmark, **GenoArmory**, targeting adversarial attacks—one of the most critical threats to GFM security. GenoArmory supports state-of-the-art attacks and defenses on GFMs, as well as providing direct access to the corresponding adversarial attack artifacts. In particular, we prioritize the following aspects in our benchmark: Our benchmark will be continuously updated to incorporate emerging attacks and defenses from the literature. Additionally, we aim to evolve the benchmark alongside the community to support newly developed methods.

3.1 GenoAdv: A dataset of adversarial examples on GFMs

An important contribution of this work is the creation of an adversarial example dataset for GFMs, named **GenoAdv**. This dataset comprises adversarial examples generated using multiple attack

methods—BertAttack [42], TextFooler [33], and FIMBA [74]—on various GFMs. While prior studies [46, 91, 49] leverage transferable adversarial examples for training, the effectiveness of such transferability remains questionable. To address this, we generate adversarial examples using diverse techniques to better capture model-specific vulnerabilities. The GenoAdv dataset offers a comprehensive and diverse set of adversarial examples across different tasks and methods, providing users with a practical resource for rapid adversarial training to enhance model robustness.

3.2 A repository of adversarial attacks artifacts

A central component of the GenoArmory benchmark is our accessible repository of adversarial attack artifacts. Given the limited availability of GFM-specific adversarial attack method—FIMBA [74] being the only one to date—we adapt existing attack techniques from language and computer vision domains to GFMs. As a result, the GenoArmory artifact repository includes adversarial examples generated by BertAttack [42], TextFooler [33], PGD [57], and FIMBA [74].

```
from GenoArmory import GenoArmory
gen = GenoArmory(model="magicslabnu/DNABERT-2-finetuned-H3",
    tokenizer="magicslabnu/DNABERT-2-finetuned-H3")
gen.get_attack_metadata(method=TextFooler,model_name=dnabert)
```

3.3 A pipeline for red-teaming GFMs

Adversarial attacks on GFMs are challenging due to variations in tokenization, architecture, configuration, and datasets, leading to inconsistent results. To address this, we propose a standardized red-teaming pipeline that includes pre-trained GFMs, datasets, hyperparameters, and adversarial examples. The pipeline integrates five state-of-the-art models—DNABERT-2 [94], Nucleotide Transformer (NT, NT2) [12], GenomeOcean [96], and HyenaDNA [64]—along with 26 DNA-based classification datasets. It provides direct access to attack artifacts Section 3.2 for standardized evaluation of adversarial robustness and supports user-defined attack methods, offering a flexible and extensible framework for evaluating model robustness.

```
import json
with open(params_file, "r") as f:
    kwargs = json.load(f)
gen.attack(attack_method='pgd', **kwargs)
```

3.4 A pipeline for evaluating defenses against adversarial attacks

In addition to efforts in developing new attack methods, researchers propose various defense strategies to counter adversarial threats. Our benchmark provides a standardized pipeline for evaluating the effectiveness of these defenses against adversarial attacks. Since no defense methods have been specifically designed for GFMs, we adapt existing state-of-the-arts from natural language and computer vision domains, i.e., adversarial training [91], ADFAR [2], and FreeLB [97], as defense baselines for GFMs. In our evaluation, we adopt existing attack methods as the base and assess the robustness of the defenses against adversarial examples generated by these attacks.

```
gen.defense(defense_method='freelb', **kwargs)
```

3.5 Reproducible evaluation framework

In addition to providing access to the attack artifacts and defense strategies, we present a standardized evaluation framework, enabling users to benchmark robustness methods. The framework includes all essential components—data loading, model training and evaluation, and accuracy-based metrics. A detailed discussion on reproducibility is provided in Appendix E.

3.6 A lightweight and easy-to-use implementation

All implementations in our framework and pipelines are built on PyTorch and Huggingface Transformers [78]. For defense evaluation, we employ the Hugging Face Trainer API to fine-tune the models. All resulting classification checkpoints are publicly available on the Hugging Face Model Hub and can be easily downloaded and applied by researchers for further studies.

3.7 A lightweight visulization framework

In our framework, we also introduce a visualization tool that enables users to explore how adversarial perturbations affect model predictions on input DNA sequences. Unlike language and computer

vision domains—where explanations often rely on heuristic attribution or prediction maps—our approach leverages genomic knowledge to validate sequence-level changes with biological expectations. Although there is a growing body of literature on explainable AI in the context of adversarial attacks [62, 13, 25, 65], these works predominantly rely on saliency-based methods. In contrast, GFMs offer a promising path forward by grounding explanations in real-world biological data and leveraging bioinformatics for more interpretable and trustworthy insights.

4 Evaluations of the Current Attacks and Defenses

In this section, we conduct a series of experiments to assess the impact of adversarial attacks and defenses on the safety of GFMs. We use DNABERT-2 [94], HyenaDNA [64], Nucleotide Transformer (NT) [12], NT2, and GenomeOcean [96] as the target models.

Models. Following Zhou et al. [94], we use DNABERT-2, NT, NT2, GenomeOcean, and HyenaDNA as target models. The first four are transformer-based models trained specifically on DNA sequences, whereas HyenaDNA utilizes a Hyena-based architecture for processing DNA sequences. We finetune all models using the sequence classification technique, following Zhou et al. [94], and utilize the finetuned models as the targets to evaluate the adversarial attacks—we generate adversarial examples that are misclassified by the target models while indistinguishable from the original examples.

		Transformer-based				Hyena-based
		DNABERT-2	NT2	NT	OG	HyenaDNA
Epigenetic Marks Prediction	H3	3	4	2	5	1
	H3K4me1	4	2	3	5	1
	H3K4me2	2	1	3	4	5
	H3K4me3	4	2	3	5	1
	H3K14ac	5	2	4	3	1
	H3K36me3	3	1	2	4	5
Epigenetic Marks Prediction	H3K9ac	4	5	2	3	1
	H3K79me3	3	2	4	5	1
	H4	3	2	5	4	1
	H4ac	5	3	2	4	1
Promoter Detection	prom_300_all	2	4	3	5	1
	prom_300_notata	1	2	4	3	5
	prom_300_tata	4	2	3	1	5
	prom_core_all	4	1	3	5	2
	prom_core_notata	2	4	5	3	1
	prom_core_tata	2	1	4	3	5
Transcription Factor Prediction (Hunan)	tf0	2	4	3	1	5
	tf1	2	4	3	1	5
	tf2	4	2	1	3	5
	tf3	1	3	2	4	5
	tf4	2	4	3	1	5
Transcription Factor Prediction (Mouse)	mouse_0	4	5	3	2	1
	mouse_1	1	4	5	3	2
	mouse_2	4	2	5	3	1
	mouse_3	2	3	1	4	5
	mouse_4	3	2	1	4	5

Figure 3: **Performance of Adversarial Attacks on Different Model Architectures.** We assess the effectiveness of the evaluated adversarial attacks across diverse model architectures, including both transformer-based models (DNABERT-2, NT, NT2, GenomeOcean) and Hyena-based model (HyenaDNA). We use the Attack Success Rate (ASR) as the primary metric to evaluate the performance of the evaluated adversarial attacks. For each experiment, we rank the top five models based on their ASR, with ranks assigned from 1 to 5. A lower rank indicates better robustness, while a higher rank reflects greater vulnerability to attacks. Our results highlight how each model performs under attack, revealing differences in vulnerability and resilience across the architectures.

Datasets. We utilize 26 datasets covering 5 tasks and 4 species, as detailed in Zhou et al. [94]. These datasets are specifically curated for genome sequence classification tasks, featuring input sequence lengths that range from 70 to 1000.

Evaluation metrics. We evaluate the effectiveness of adversarial attacks using the Attack Success Rate (ASR) and assess defense strategies using the Defense Success Rate (DSR) as detailed in Appendix I.2. Accuracy is used as the core metric to quantify the impact of both attacks and defenses.

Table 1: **Adversarial Attack Performance of the Evaluated Method.** We conduct experiments to assess the effectiveness of the evaluated attack method against adversarial attacks. The table presents a comparison of target model performance before and after applying the evaluated attack. We report Attack Success Rate (ASR) as the primary evaluation metric, with variance omitted as they are all $\leq 2\%$. The final columns present the average Attack Success Rate (ASR) across all GFM models for each specific attack. The last row similarly shows the average ASR across all attacks for each specific GFM. Additionally, for each attack, individual ASR scores are ranked from highest to lowest, with the rank displayed in brackets next to the score.

	Transformer-based				Hyena-based	
Attack	DNABERT-2	NT	NT2	GenomeOcean	HyenaDNA	Avg
BertAttack	96.23%(5)	99.87%(1)	99.56%(4)	99.57%(3)	99.75%(2)	99.00%
TextFooler	92.37%(4)	96.69%(2)	96.56%(3)	99.54%(1)	88.45%(5)	94.72%
PGD	38.28%(2)	38.23%(3)	34.41%(5)	36.57%(4)	47.94%(1)	39.09%
FIMBA	39.94%(2)	37.66%(3)	36.50%(4)	41.06%(1)	30.35%(5)	37.10%
Attack ASR	66.71% (3.25)	68.11% (2.25)	66.76% (4)	69.19% (2.25)	66.62% (3.25)	

4.1 Evaluating adversarial attacks on GFMs

We utilize the same datasets and models as described in Section 3.2 to ensure consistency in our evaluation. We conduct each evaluation three times with different random seeds and present the average and standard deviation for each metric.

Baseline attack artifacts. We test four baseline attack methods—BertAttack [42], TextFooler [33], PGD [57], and FIMBA [74]—to assess their effectiveness in generating adversarial examples. Experiments are conducted on 5 GFMs, covering both transformer-based and Hyena-based architectures, with implementation details provided in Appendix I.3. Attack performance is primarily measured using ASR, and methods are ranked based on their average ASR across all datasets.

Results. In Figure 3 and Table 1, our results highlight the effectiveness of the evaluated attacks in generating adversarial examples that are misclassified by target models. We have below observations.

- GenomeOcean exhibits greater susceptibility to adversarial attacks than classification models (DNABERT-2, NT2), as evidenced by higher ASR and ranks across all GFMs. This observation aligns with the findings in Ebrahimi et al. [17], Wang et al. [75].
- NT2 demonstrates the highest robustness, indicated by its lowest average rank, potentially due to its use of BPE tokenization. GFMs employing BPE tokenization (DNABERT-2, NT2) appear to be more robust than those using k-mer tokenization (NT). BPE's subword structure allows for partial token retention despite alterations, hindering significant semantic or biological shifts. Interestingly, while NT2's average ASR is higher than HyenaDNA's (the lowest overall), its ASR rank is lower. In contrast, NT shares the highest ASR rank with GenomeOcean but has a lower ASR. The discrepancy stems from NT consistently achieving high ASR across all attacks, while GenomeOcean performs best on TextFooler and FIMBA but poorly on BertAttack and PGD.
- BertAttack yields the highest average ASR across GFMs, while FIMBA, the only genome-specific attack, shows the lowest, indicating limited effectiveness. This ineffectiveness may be due to constraints in the released FIMBA code [1] and evaluation setup in Skovorodnikov and Alkhzaimi [74]. However, traditional NLP-based adversarial attacks such as BertAttack and TextFooler already achieve a high ASR in these models. This underscores the importance of developing defense mechanisms tailored for GFM tasks to ensure their safety.

4.2 Evaluating adversarial defenses

Each experiment is repeated three times with different random seeds on the same datasets and models, and we report the mean and standard deviation of each evaluation metric.

Baseline defenses. We assess the robustness of five GFM models against adversarial attacks using three defense baselines: adversarial training [91] (employing TextFooler for data augmentation), FreeLB [97], and ADFAR [2]. Defenses were evaluated against BertAttack, TextFooler, and PGD attacks, with the DSR as the primary robustness metric.

[1] https://github.com/HeorhiiS/fimba-attack

Table 2: **Defense Performance Under Adversarial Attacks.** We conducted experiments to evaluate the performance of a defense method against adversarial attacks. The table compares the performance of target models, both with and without the evaluated defense, under BertAttack, TextFooler, and PGD attacks. The Defense Success Rate (DSR) is used as the primary evaluation metric, with variance omitted as they are all $\leq 2\%$. The best DSR values are highlighted in bold. In the table, **AT** denotes traditional adversarial training. We observe that ADFAR is the most effective defense based on DSR, particularly against BertAttack and TextFooler.

		Transformer-based				Hyena-based
Attack Method	Defense	DNABERT-2	NT	NT2	GenomeOcean	HyenaDNA
BertAttack	N/A	3.77%	0.13%	0.44%	0.43%	0.25%
	AT	4.06%	0.21%	0.46%	0.60%	0.81%
	FreeLB	4.34%	0.67%	0.71%	**2.94%**	1.12%
	ADFAR	**21.84%**	**4.95%**	**6.96%**	1.18%	**1.50%**
PGD	N/A	61.73%	61.77%	65.59%	63.43%	52.06%
	AT	**64.92%**	79.10%	82.02%	**66.14%**	85.67%
	FreeLB	64.07%	**79.38%**	**88.53%**	65.96%	**86.99%**
	ADFAR	63.48%	63.44%	72.89%	65.87%	83.74%
TextFooler	N/A	7.63%	3.31%	3.44%	0.46%	11.55%
	AT	20.97%	42.88%	18.95%	18.51%	**84.19%**
	FreeLB	18.39%	42.94%	18.16%	17.33%	69.56%
	ADFAR	**32.88%**	**67.07%**	**22.00%**	**46.18%**	80.82%

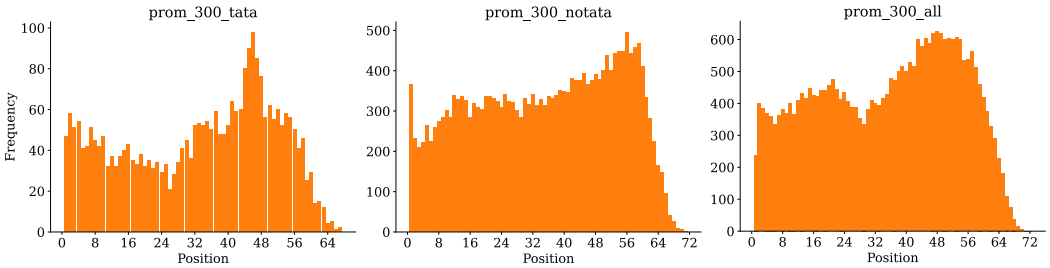

Figure 4: **Examples of the visualization of GFMs with adversarial attacks.** We present the results of the three tasks of the DNABERT-2 model under BertAttack. All subsequence changes occur at the subword tokenizer level using Byte Pair Encoding (BPE) [71]. The visualization highlights which parts of the sequence are most significant for the model's classification performance. Specifically, we present the frequency with which the adversarial attack modifies the sequence. A higher frequency indicates that the subsequence is more critical for the model's ability to perform classification tasks.

Results. As shown in Table 2, we have below observations:

- ADFAR achieves the highest overall DSR, significantly outperforming other defenses against BertAttack and TextFooler. However, ADFAR performs poorly against the PGD attack.
- FreeLB obtains better DSR against PGD, possibly due to it smooths the adversarial loss during training, which somewhat improves robustness.
- AT is less effective than ADFAR and FreeLB against BertAttack and TextFooler, although AT performs comparably to FreeLB against PGD attacks.
- While the model architecture does not significantly affect overall defense performance, specific models show distinct advantages, e.g., DNABERT-2 and NT2 show a greater defense improvement against BertAttack, while HyenaDNA demonstrates a better defense against TextFooler and PGD.

4.3 Visualization of adversarial attacks

In this experiment, we visualize adversarial attacks on target models with our framework. We utilize BertAttack to generate adversarial examples and visualize the results using the DNABERT-2 model. The visualization highlights the subsequences that are most significant for the model's classification performance, specifically focusing on the frequency with which the adversarial attack modifies the

Table 3: **Defense Performance Augmented with the GenoAdv Dataset.** We conduct experiments to evaluate the performance of a model augmented with the GenoAdv dataset against adversarial attacks. The table compares the performance of the target model, both with and without the GenoAdv dataset augmentation, under BertAttack, TextFooler, and PGD attacks. We report ASR as the primary evaluation metric, with variance omitted as they are all $\leq 2\%$. The best results are highlighted in bold. In the table, **AT** denotes traditional adversarial training. We observe that GenoAdv samples are more effective than TextFooler samples under traditional adversarial training methods.

		Transformer-based				Hyena-based
Attack Method	Defense	DNABERT-2	NT	NT2	GenomeOcean	HyenaDNA
BertAttack	N/A	3.77%	0.13%	0.44%	0.43%	0.25%
	AT	4.06%	0.21%	0.46%	0.60%	0.81%
	GenoAdv	**5.17%**	**0.69%**	**0.59%**	**0.73%**	**5.23%**
PGD	N/A	61.73%	61.77%	65.59%	63.43%	52.06%
	AT	64.92%	79.10%	**82.02%**	66.14%	**85.67%**
	GenoAdv	**69.32%**	**79.31%**	75.57%	**67.10%**	84.52%
TextFooler	N/A	7.63%	3.31%	3.44%	0.46%	11.55%
	AT	20.97%	42.88%	18.95%	18.51%	**84.19%**
	GenoAdv	**22.19%**	**44.05%**	**20.56%**	**19.45%**	81.99%

sequence. We present the frequency of subsequence changes at the subword tokenizer level using Byte Pair Encoding (BPE). As shown in Figure 4, the visualization is generated by analyzing the frequency of subsequence changes across all datasets and models, providing insight into the most critical subsequences for the model's classification performance.

4.4 Performance of model augmented with GenoAdv dataset

In order to show the effectiveness of the GenoAdv dataset, we conduct experiments to evaluate the performance of the model augmented with the GenoAdv dataset. We use BertAttack, TextFooler, and PGD to evaluate the DSR on 5 GFMs. In our experiment, we perform traditional adversarial training with TextFooler-augmented data as a baseline, and compare it to the same training approach using the GenoAdv dataset. We conduct each evaluation three times with different random seeds and present the average and standard deviation for each metric.

Results: As shown in Table 3, adversarial training with GenoAdv data yields stronger robustness against adversarial attacks compared to training with only TextFooler-augmented samples in most cases. This suggests that the GenoAdv dataset offers valuable augmentation data to mitigate the vulnerability of GFMs. Specifically, using GenoAdv data to do data augmentation leads to a performance improvement of 34.71% over TextFooler-based adversarial training.

4.5 Quantization influence on adversarial attacks

To evaluate the influence of quantization on evaluated attacks, we conduct experiments on quantized versions of target models. Inside those quantization methods, some of them are based on the traditional quantization methods, such as uniform quantization, and some of them are based on the outluer-removal quantization methods, such as OutEffHop [28]. Following the quantization setup in Luo et al. [55] and Wu et al. [80], we evaluate the performance of the attacks on quantized models with 8-bit weights and 8-bit activations (W8A8), comparing them to the original models to analyze the impact of quantization on attack detectability.

Results. In Table 4, our results highlight the effectiveness of quantization in improving the robustness of target models against adversarial attacks. Specifically, we observe that the evaluated attacks achieve a lower ASR on quantized models compared to the original models, indicating that quantization strengthens the defenses against these attacks. Additionally, the outlier-free quantization method also reduces the ASR of the evaluated attacks. This outcome suggests that quantization can improve model robustness against adversarial attacks. One possible explanation is that quantization introduces "flat regions" in the loss landscape, which diminishes the model's sensitivity to small perturbations. This observation aligns with the findings reported in Lin et al. [48].

However, we find that the OutEffHop quantization method results in a higher ASR compared to traditional quantization methods, indicating that outlier-removal quantization can compromise the

Table 4: **Performance of the evaluated attacks on quantized models.** We perform experiments to assess how quantization affects the effectiveness of adversarial attacks on target models. The table compares model performance before and after quantization under BertAttack and TextFooler attacks. Attack Success Rate (ASR) serves as the primary evaluation metric, with variance omitted as they are all $\leq 2\%$. The best results are highlighted in bold.

Attack Method	Model	Quantized Method	ASR (↓)
BertAttack	DNABERT-2	-	96.23
		Vanilla	**59.46**
		OutEffHop	64.71
	NT1	-	99.87
		Vanilla	**99.37**
		OutEffHop	99.42
TextFooler	DNABERT-2	-	92.37
		Vanilla	**19.90**
		OutEffHop	21.34
	NT1	-	98.23
		Vanilla	**66.57**
		OutEffHop	68.53

robustness of target models against adversarial attacks. A possible reason for this is that the OutEffHop method removes outliers in the model's attention architecture, which improves the quantization process. However, this improvement also eliminates the "flat regions" in the loss landscape that are critical to the robustness provided by traditional quantization methods. We also find that quantization significantly impacts DNABERT-2 models, but has minimal effect on NT1 models, suggesting model-specific robustness gains. Notably, TextFooler is more affected by quantization than BERT-Attack, likely due to its dependence on precise word importance scores and synonym substitutions, which are disrupted by quantization-induced shifts in decision boundaries.

5 Discussion and Conclusion

We introduce GenoArmory, the first unified adversarial attack benchmark for DNA-based Genomic Foundation Models (GFMs). Our benchmark offers an accessible, reproducible, and comprehensive framework, enabling users to confidently evaluate and compare adversarial robustness in GFMs. Also, to encourage broad participation, we do not restrict the architectures of threat or target models. Instead, GenoArmory offers a standardised framework for evaluating adversarial attacks and defenses, with periodic updates to incorporate state-of-the-art methods in the field. Methodologically, compared to adversarial attack benchmarks in language and computer vision [92, 11, 15], GenoArmory includes visualization tools that facilitate deeper insights into the evaluated attacks—leveraging the fact that GFM data is inherently structured and scientifically meaningful.

Limitations. Although GenoArmory provides a comprehensive evaluation of adversarial attacks and defenses on DNA-based GFMs, it still has several limitations. For example, GenoArmory currently excludes RNA-based GFMs and is limited to classification tasks, leaving other task types and modalities unaddressed.

Developing a comprehensive benchmark is essential, as GFM safety is often underestimated. Yet, insufficient safeguards hinder their advancement and pose risks to scientific progress. A key challenge in improving GFM safety is the lack of a comprehensive benchmark for evaluating vulnerabilities. In this paper, we provide the **first** in-depth analysis of DNA-based attacks on leading GFMs using such a benchmark. However, this serves only as a foundation—future work must extend it to include broader attack vectors, such as RNA-based model attacks, to ensure more robust evaluation. Greater focus is also needed on generative GFMs, such as Evo [63], which remain underrepresented in safety evaluations. Beyond benchmarks, the lack of automated tools for assessing the safety of generated genomic sequences—unlike in image or speech domains—poses a critical gap. This highlights the urgent need for robust, domain-specific evaluation frameworks to ensure safe and ethical deployment of GFMs.

Automatic sequence data judgment system provides a framework for assessing sequence differences to evaluate the safety of generated genomic sequences. Prior work on sequence functionality [73, 22] and ortholog analysis [29] demonstrates that ortholog comparisons can reveal relationships between genomic sequences, informing safety assessments. Building on this idea, Emms and Kelly [19] introduce a method to calculate ortholog differences within genomic sequences. By using the distance between sequence orthologs, researchers can quantify differences between generated sequences and known harmful genomic sequences, providing a method to assess sequence safety. This approach enables the development of an automated system for sequence evaluation, improving efficiency in safety assessments. Additionally, leveraging large language models (LLMs) like Qwen [9] and Llama3 [16] to generate genomic sequences enhances the model's diversity and robustness.

References

[1] Armen Aghajanyan, Akshat Shrivastava, Anchit Gupta, Naman Goyal, Luke Zettlemoyer, and Sonal Gupta. Better fine-tuning by reducing representational collapse. *arXiv preprint arXiv:2008.03156*, 2020.

[2] Rongzhou Bao, Jiayi Wang, and Hai Zhao. Defending pre-trained language models from adversarial word substitution without performance sacrifice. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3248–3258, Online, August 2021. Association for Computational Linguistics.

[3] Gonzalo Benegas, Sanjit Singh Batra, and Yun S Song. Dna language models are powerful predictors of genome-wide variant effects. *Proceedings of the National Academy of Sciences*, 120(44):e2311219120, 2023.

[4] Nicholas Carlini and David Wagner. Defensive distillation is not robust to adversarial examples. *arXiv preprint arXiv:1607.04311*, 2016.

[5] Nicholas Carlini and David Wagner. Towards Evaluating the Robustness of Neural Networks . In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 39–57, Los Alamitos, CA, USA, May 2017. IEEE Computer Society.

[6] Pin-Yu Chen, Yash Sharma, Huan Zhang, Jinfeng Yi, and Cho-Jui Hsieh. Ead: elastic-net attacks to deep neural networks via adversarial examples. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[7] Yong Cheng, Lu Jiang, and Wolfgang Macherey. Robust neural machine translation with doubly adversarial inputs. In Anna Korhonen, David Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4324–4333, Florence, Italy, July 2019. Association for Computational Linguistics.

[8] Davide Chicco and Giuseppe Jurman. The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation. *BMC genomics*, 21:1–13, 2020.

[9] Yunfei Chu, Jin Xu, Xiaohuan Zhou, Qian Yang, Shiliang Zhang, Zhijie Yan, Chang Zhou, and Jingren Zhou. Qwen-audio: Advancing universal audio understanding via unified large-scale audio-language models. *arXiv preprint arXiv:2311.07919*, 2023.

[10] Gilad Cohen, Guillermo Sapiro, and Raja Giryes. Detecting adversarial samples using influence functions and nearest neighbors. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14453–14462, 2020.

[11] Francesco Croce, Maksym Andriushchenko, Vikash Sehwag, Edoardo Debenedetti, Nicolas Flammarion, Mung Chiang, Prateek Mittal, and Matthias Hein. Robustbench: a standardized adversarial robustness benchmark. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.

[12] 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*, pages 1–11, 2024.

[13] Prathyusha Devabhakthini, Sasmita Parida, Raj Mani Shukla, and Suvendu Chandan Nayak. Analyzing the impact of adversarial examples on explainable machine learning. *arXiv preprint arXiv:2307.08327*, 2023.

[14] Tommaso Di Noia, Daniele Malitesta, and Felice Antonio Merra. Taamr: Targeted adversarial attack against multimedia recommender systems. In *2020 50th Annual IEEE/IFIP international conference on dependable systems and networks workshops (DSN-W)*, pages 1–8. IEEE, 2020.

[15] Yinpeng Dong, Qi-An Fu, Xiao Yang, Tianyu Pang, Hang Su, Zihao Xiao, and Jun Zhu. Benchmarking adversarial robustness on image classification. In *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 321–331, 2020.

[16] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[17] Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. HotFlip: White-box adversarial examples for text classification. In Iryna Gurevych and Yusuke Miyao, editors, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 31–36, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[18] Tahir Elgamrani, Reda Elgaf, and Yousra Chtouki. Adversarial attack defense techniques: A study of defensive distillation and adversarial re-training on cifar-10 and mnist. In *2024 International Conference on Computer and Applications (ICCA)*, pages 1–4. IEEE, 2024.

[19] David M Emms and Steven Kelly. Orthofinder: phylogenetic orthology inference for comparative genomics. *Genome biology*, 20:1–14, 2019.

[20] Reuben Feinman, Ryan R. Curtin, Saurabh Shintre, and Andrew B. Gardner. Detecting adversarial samples from artifacts, 2017.

[21] Veniamin Fishman, Yuri Kuratov, Aleksei Shmelev, Maxim Petrov, Dmitry Penzar, Denis Shepelin, Nikolay Chekanov, Olga Kardymon, and Mikhail Burtsev. Gena-lm: a family of open-source foundational dna language models for long sequences. *Nucleic Acids Research*, 53 (2):gkae1310, 2025.

[22] Sarah E Flanagan, Ann-Marie Patch, and Sian Ellard. Using sift and polyphen to predict loss-of-function and gain-of-function mutations. *Genetic testing and molecular biomarkers*, 14 (4):533–537, 2010.

[23] Xi Fu, Shentong Mo, Alejandro Buendia, Anouchka P Laurent, Anqi Shao, Maria del Mar Alvarez-Torres, Tianji Yu, Jimin Tan, Jiayu Su, Romella Sagatelian, et al. A foundation model of transcription across human cell types. *Nature*, pages 1–9, 2025.

[24] Ji Gao, Jack Lanchantin, Mary Lou Soffa, and Yanjun Qi. Black-box generation of adversarial text sequences to evade deep learning classifiers. In *2018 IEEE Security and Privacy Workshops (SPW)*, pages 50–56. IEEE, 2018.

[25] Rokas Gipiškis, Diletta Chiaro, Marco Preziosi, Edoardo Prezioso, and Francesco Piccialli. The impact of adversarial attacks on interpretable semantic segmentation in cyber–physical systems. *IEEE Systems Journal*, 17(4):5327–5334, 2023.

[26] Shreya Goyal, Sumanth Doddapaneni, Mitesh M Khapra, and Balaraman Ravindran. A survey of adversarial defenses and robustness in nlp. *ACM Computing Surveys*, 55(14s):1–39, 2023.

[27] Katarína Grešová, Vlastimil Martinek, David Čechák, Petr Šimeček, and Panagiotis Alexiou. Genomic benchmarks: a collection of datasets for genomic sequence classification. *BMC Genomic Data*, 24(1):25, 2023.

[28] Jerry Yao-Chieh Hu, Pei-Hsuan Chang, Haozheng Luo, Hong-Yu Chen, Weijian Li, Wei-Po Wang, and Han Liu. Outlier-efficient hopfield layers for large transformer-based models. In *ICML*, 2024.

[29] Roy A Jensen. Orthologs and paralogs-we need to get it right. *Genome biology*, 2(8): interactions1002–1, 2001.

[30] Yanrong Ji, Zhihan Zhou, Han Liu, and Ramana V Davuluri. DNABERT: pre-trained Bidirectional Encoder Representations from Transformers model for DNA-language in genome. 2021.

[31] Robin Jia, Aditi Raghunathan, Kerem Göksel, and Percy Liang. Certified robustness to adversarial word substitutions. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4129–4142, Hong Kong, China, November 2019. Association for Computational Linguistics.

[32] Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Tuo Zhao. SMART: Robust and efficient fine-tuning for pre-trained natural language models through principled regularized optimization. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2177–2190, Online, July 2020. Association for Computational Linguistics.

[33] Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. Is bert really robust? a strong baseline for natural language attack on text classification and entailment. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 8018–8025, 2020.

[34] Kaiwen Jin, Yifeng Xiong, Shuya Lou, and Zhen Yu. Mafd: Multiple adversarial features detector for enhanced detection of text-based adversarial examples. *Neural Processing Letters*, 56(6):251, December 2024. ISSN 1573-773X. doi: 10.1007/s11063-024-11710-0.

[35] Anowarul Kabir, Manish Bhattarai, Selma Peterson, Yonatan Najman-Licht, Kim Ø Rasmussen, Amarda Shehu, Alan R Bishop, Boian Alexandrov, and Anny Usheva. Dna breathing integration with deep learning foundational model advances genome-wide binding prediction of human transcription factors. *Nucleic Acids Research*, 52(19):e91–e91, 2024.

[36] Zong Ke, Shicheng Zhou, Yining Zhou, Chia Hong Chang, and Rong Zhang. Detection of ai deepfake and fraud in online payments using gan-based models. *arXiv preprint arXiv:2501.07033*, 2025.

[37] Valentin Khrulkov and Ivan Oseledets. Geometry score: A method for comparing generative adversarial networks, 2018.

[38] Alexey Kurakin, Ian J Goodfellow, and Samy Bengio. Adversarial examples in the physical world. In *Artificial intelligence safety and security*, pages 99–112. Chapman and Hall/CRC, 2018.

[39] Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. *Advances in neural information processing systems*, 31, 2018.

[40] Jinfeng Li, Shouling Ji, Tianyu Du, Bo Li, and Ting Wang. Textbugger: Generating adversarial text against real-world applications. In *Proceedings 2019 Network and Distributed System Security Symposium*, NDSS 2019. Internet Society, 2019.

[41] Linyang Li and Xipeng Qiu. Token-aware virtual adversarial training in natural language understanding. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(9):8410–8418, May 2021.

[42] Linyang Li, Ruotian Ma, Qipeng Guo, Xiangyang Xue, and Xipeng Qiu. BERT-ATTACK: Adversarial attack against BERT using BERT. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6193–6202, Online, November 2020. Association for Computational Linguistics.

[43] Linyang Li, Demin Song, and Xipeng Qiu. Text adversarial purification as defense against adversarial attacks. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 338–350, Toronto, Canada, July 2023. Association for Computational Linguistics.

[44] Maosen Li, Cheng Deng, Tengjiao Li, Junchi Yan, Xinbo Gao, and Heng Huang. Towards transferable targeted attack. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 641–649, 2020.

[45] Tianhao Li, Jingyu Lu, Chuangxin Chu, Tianyu Zeng, Yujia Zheng, Mei Li, Haotian Huang, Bin Wu, Zuoxian Liu, Kai Ma, Xuejing Yuan, Xingkai Wang, Keyan Ding, Huajun Chen, and Qiang Zhang. Scisafeeval: A comprehensive benchmark for safety alignment of large language models in scientific tasks, 2024.

[46] Yingwei Li, Song Bai, Yuyin Zhou, Cihang Xie, Zhishuai Zhang, and Alan Yuille. Learning transferable adversarial examples via ghost networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 11458–11465, 2020.

[47] Zongyi Li, Jianhan Xu, Jiehang Zeng, Linyang Li, Xiaoqing Zheng, Qi Zhang, Kai-Wei Chang, and Cho-Jui Hsieh. Searching for an effective defender: Benchmarking defense against adversarial word substitution. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3137–3147, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

[48] Ji Lin, Chuang Gan, and Song Han. Defensive quantization: When efficiency meets robustness. In *International Conference on Learning Representations*, 2019.

[49] Hong Liu, Mingsheng Long, Jianmin Wang, and Michael Jordan. Transferable adversarial training: A general approach to adapting deep classifiers. In *International conference on machine learning*, pages 4013–4022. PMLR, 2019.

[50] Qin Liu, Rui Zheng, Bao Rong, Jingyi Liu, ZhiHua Liu, Zhanzhan Cheng, Liang Qiao, Tao Gui, Qi Zhang, and Xuanjing Huang. Flooding-X: Improving BERT‘s resistance to adversarial attacks via loss-restricted fine-tuning. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5634–5644, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[51] Yujie Liu, Shuai Mao, Xiang Mei, Tao Yang, and Xuran Zhao. Sensitivity of adversarial perturbation in fast gradient sign method. In *2019 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 433–436, 2019.

[52] Zicheng Liu, Jiahui Li, Lei Xin, Siyuan Li, Chang Yu, Zelin Zang, Cheng Tan, Yufei Huang, yajingbai, Jun Xia, and Stan Z. Li. Genebench: Systematic evaluation of genomic foundation models and beyond, 2025.

[53] I Loshchilov. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[54] Haozheng Luo, Jiahao Yu, Wenxin Zhang, Jialong Li, Jerry Yao-Chieh Hu, Xinyu Xing, and Han Liu. Decoupled alignment for robust plug-and-play adaptation, 2024.

[55] Haozheng Luo, Chenghao Qiu, Maojiang Su, Zhihan Zhou, Zoe Mehta, Guo Ye, Jerry Yao-Chieh Hu, and Han Liu. Fast and low-cost genomic foundation models via outlier removal. *arXiv preprint arXiv:2505.00598*, 2025.

[56] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018.

[57] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018.

[58] Daniel Mas Montserrat and Alexander G Ioannidis. Adversarial attacks on genotype sequences. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2023.

[59] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2574–2582, 2016.

[60] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Universal adversarial perturbations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1765–1773, 2017.

[61] Konda Reddy Mopuri, Utkarsh Ojha, Utsav Garg, and R. Venkatesh Babu. Nag: Network for adversary generation, 2018.

[62] Ofir Moshe, Gil Fidel, Ron Bitton, and Asaf Shabtai. Improving interpretability via regularization of neural activation sensitivity. *Machine Learning*, 113(9):6165–6196, 2024.

[63] 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.

[64] 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, 2024.

[65] Utku Ozbulak, Baptist Vandersmissen, Azarakhsh Jalalvand, Ivo Couckuyt, Arnout Van Messem, and Wesley De Neve. Investigating the significance of adversarial attacks and their relation to interpretability for radar-based human activity recognition systems. *Computer Vision and Image Understanding*, 202:103111, 2021.

[66] Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *2016 IEEE symposium on security and privacy (SP)*, pages 582–597. IEEE, 2016.

[67] Aidan Peppin, Anka Reuel, Stephen Casper, Elliot Jones, Andrew Strait, Usman Anwar, Anurag Agrawal, Sayash Kapoor, Sanmi Koyejo, Marie Pellat, et al. The reality of ai and biorisk. *arXiv preprint arXiv:2412.01946*, 2024.

[68] Omid Poursaeed, Isay Katsman, Bicheng Gao, and Serge Belongie. Generative adversarial perturbations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4422–4431, 2018.

[69] Fanchao Qi, Yangyi Chen, Mukai Li, Yuan Yao, Zhiyuan Liu, and Maosong Sun. ONION: A simple and effective defense against textual backdoor attacks. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9558–9566, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

[70] Mohammad Amaan Sayeed, Hanan Aldarmaki, and Boulbaba Ben Amor. Gene pathogenicity prediction using genomic foundation models. In *AAAI 2024 Spring Symposium on Clinical Foundation Models*, 2024.

[71] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In Katrin Erk and Noah A. Smith, editors, *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany, August 2016. Association for Computational Linguistics.

[72] Erfan Shayegani, Md Abdullah Al Mamun, Yu Fu, Pedram Zaree, Yue Dong, and Nael Abu-Ghazaleh. Survey of vulnerabilities in large language models revealed by adversarial attacks, 2023.

[73] Ngak-Leng Sim, Prateek Kumar, Jing Hu, Steven Henikoff, Georg Schneider, and Pauline C Ng. Sift web server: predicting effects of amino acid substitutions on proteins. *Nucleic acids research*, 40(W1):W452–W457, 2012.

[74] Heorhii Skovorodnikov and Hoda Alkhzaimi. Fimba: Evaluating the robustness of ai in genomics via feature importance adversarial attacks. *arXiv preprint arXiv:2401.10657*, 2024.

[75] Jindong Wang, Xixu HU, Wenxin Hou, Hao Chen, Runkai Zheng, Yidong Wang, Linyi Yang, Wei Ye, Haojun Huang, Xiubo Geng, Binxing Jiao, Yue Zhang, and Xing Xie. On the robustness of chatGPT: An adversarial and out-of-distribution perspective. In *ICLR 2023 Workshop on Trustworthy and Reliable Large-Scale Machine Learning Models*, 2023.

[76] Jingyi Wang, Guoliang Dong, Jun Sun, Xinyu Wang, and Peixin Zhang. Adversarial sample detection for deep neural network through model mutation testing. In *2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE)*, pages 1245–1256. IEEE, 2019.

[77] Rey Wiyatno and Anqi Xu. Maximal jacobian-based saliency map attack, 2018.

[78] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, October 2020. Association for Computational Linguistics.

[79] Aming Wu, Yahong Han, Quanxin Zhang, and Xiaohui Kuang. Untargeted adversarial attack via expanding the semantic gap. In *2019 IEEE International Conference on Multimedia and Expo (ICME)*, pages 514–519. IEEE, 2019.

[80] Shang Wu, Yen-Ju Lu, Haozheng Luo, Maojiang Su, Jerry Yao-Chieh Hu, Jiayi Wang, Jing Liu, Najim Dehak, Jesus Villalba, and Han Liu. SPARQ: Outlier-free speechLM with fast adaptation and robust quantization, 2025.

[81] Hao Xuan, Bokai Yang, and Xingyu Li. Exploring the impact of temperature scaling in softmax for classification and adversarial robustness, 2025.

[82] Heng Yang and Ke Li. Omnigenome: Aligning rna sequences with secondary structures in genomic foundation models. *arXiv preprint arXiv:2407.11242*, 2024.

[83] Yahan Yang, Soham Dan, Dan Roth, and Insup Lee. In and out-of-domain text adversarial robustness via label smoothing. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 657–669, Toronto, Canada, July 2023. Association for Computational Linguistics.

[84] Mao Ye, Chengyue Gong, and Qiang Liu. SAFER: A structure-free approach for certified robustness to adversarial word substitutions. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3465–3475, Online, July 2020. Association for Computational Linguistics.

[85] Peng Ye, Weiqiang Bai, Yuchen Ren, Wenran Li, Lifeng Qiao, Chaoqi Liang, Linxiao Wang, Yuchen Cai, Jianle Sun, Zejun Yang, et al. Genomics-fm: Universal foundation model for versatile and data-efficient functional genomic analysis. *bioRxiv*, pages 2024–07, 2024.

[86] Zhixing Ye, Xinwen Cheng, and Xiaolin Huang. Fg-uap: Feature-gathering universal adversarial perturbation. In *2023 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2023.

[87] Jiahao Yu, Haozheng Luo, Jerry Yao-Chieh Hu, Wenbo Guo, Han Liu, and Xinyu Xing. BOOST: Enhanced jailbreak of large language model via slient eos tokens, 2025.

[88] Jiehang Zeng, Jianhan Xu, Xiaoqing Zheng, and Xuanjing Huang. Certified robustness to text adversarial attacks by randomized [mask]. *Computational Linguistics*, 49(2):395–427, 06 2023. ISSN 0891-2017.

[89] Chaoning Zhang, Philipp Benz, Adil Karjauv, and In So Kweon. Data-free universal adversarial perturbation and black-box attack. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 7868–7877, 2021.

[90] Jiebao Zhang, Wenhua Qian, Jinde Cao, and Dan Xu. Lp-bfgs attack: An adversarial attack based on the hessian with limited pixels. *Computers & Security*, 140:103746, 2024.

[91] Haizhong Zheng, Ziqi Zhang, Juncheng Gu, Honglak Lee, and Atul Prakash. Efficient adversarial training with transferable adversarial examples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1181–1190, 2020.

[92] Meixi Zheng, Xuanchen Yan, Zihao Zhu, Hongrui Chen, and Baoyuan Wu. Blackboxbench: A comprehensive benchmark of black-box adversarial attacks. *arXiv preprint arXiv:2312.16979*, 2023.

[93] Rui Zheng, Shihan Dou, Yuhao Zhou, Qin Liu, Tao Gui, Qi Zhang, Zhongyu Wei, Xuanjing Huang, and Menghan Zhang. Detecting adversarial samples through sharpness of loss landscape. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Findings of the Association for Computational Linguistics: ACL 2023*, pages 11282–11298, Toronto, Canada, July 2023. Association for Computational Linguistics.

[94] Zhihan Zhou, Yanrong Ji, Weijian Li, Pratik Dutta, Ramana V Davuluri, and Han Liu. DNABERT-2: Efficient foundation model and benchmark for multi-species genomes. In *The Twelfth International Conference on Learning Representations*, 2024.

[95] Zhihan Zhou, Weimin Wu, Harrison Ho, Jiayi Wang, Lizhen Shi, Ramana V Davuluri, Zhong Wang, and Han Liu. DNABERT-s: Pioneering species differentiation with species-aware DNA embeddings, 2025.

[96] Zhihan Zhou, Weimin Wu, Jieke Wu, Lizhen Shi, Zhong Wang, and Han Liu. Genomeocean: Efficient foundation model for genome generation, 2025.

[97] Chen Zhu, Yu Cheng, Zhe Gan, Siqi Sun, Tom Goldstein, and Jingjing Liu. Freelb: Enhanced adversarial training for natural language understanding. In *International Conference on Learning Representations*, 2020.

Supplement Material

A Open Science

We release the code, pretrained checkpoints, and datasets used in our work. The code is available at this GitHub repository, and the pretrained checkpoints are hosted on HuggingFace. The GenoAdv dataset is hosted on Hugging Face Datasets and can be accessed directly through their platform.

B Boarder Impact

This paper seeks to advance the trustworthiness of genomic foundation models (GFMs). While the work does not have immediate social implications, it represents a step toward creating more reliable GFMs. However, the adversarial samples released in the **GenoAdv** dataset and experiments can provide incorrect classification for existing GFMs.

C Related Work

In this section, we explore the background of vulnerabilities in GFMs. We begin by introducing benchmarks for evaluating adversarial attacks on GFMs, including standard datasets, metrics, and evaluation protocols. Next, we review existing adversarial attack methods tailored for GFMs, such as BERT-Attack [42] and PGD [57]. Finally, we discuss defense strategies against these attacks, covering approaches like FreeLB [97] and ADFAR [2].

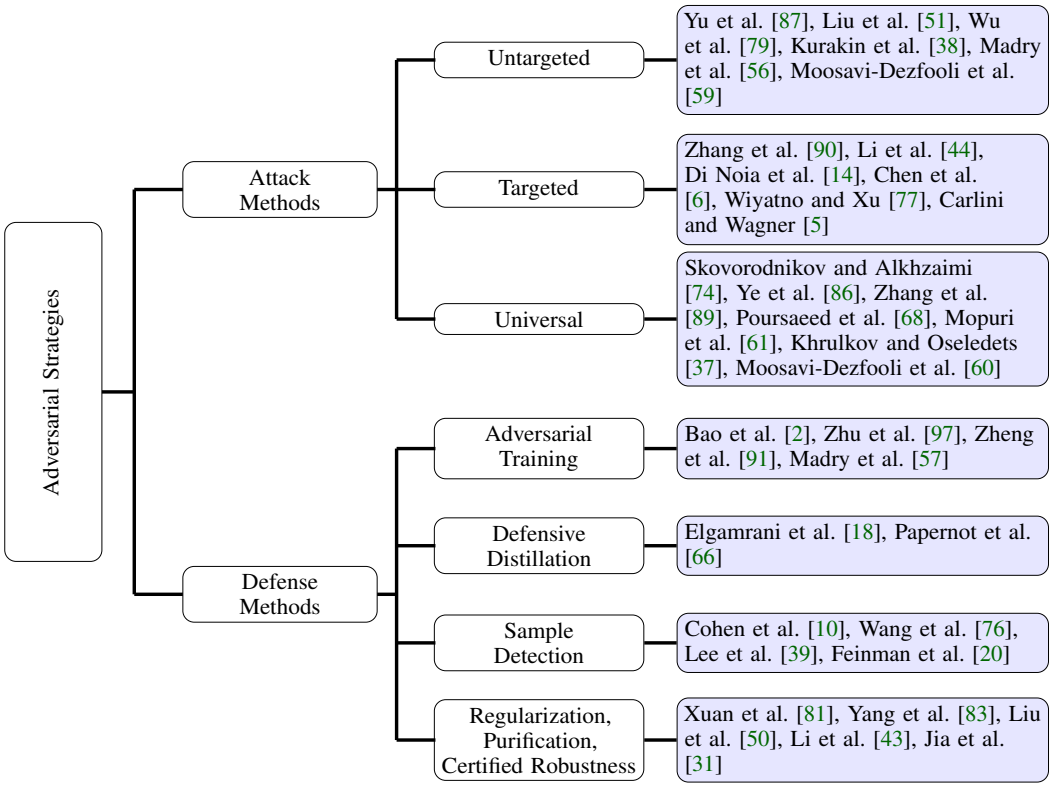

Figure 5: **Taxonomy of Adversarial Strategies.**

C.1 Benchmarks

The GUE benchmark [94] encompasses a variety of genome classification tasks, including promoter detection, transcription factor prediction, and COVID variant classification. These tasks are designed to assess model performance across multiple species, such as humans, fungi, viruses, and yeast. Building on this, GUE+ extends the benchmark to focus on tasks involving longer input sequences, ranging from 5000 to 10000 base pairs, to evaluate models' capabilities in processing and analyzing complex genomic data. The GUE benchmark assesses model performance using metrics such as Accuracy, F1-score, and Matthews Correlation Coefficient (MCC) [8].

Meanwhile, GenBench [52] is a comprehensive benchmarking suite tailored for evaluating the performance of GFMs. It systematically analyzes datasets from diverse biological domains, with a focus on both short-range and long-range genomic tasks. These tasks encompass essential areas such as coding regions, non-coding regions, and genome structure. For classification tasks, GenBench uses cross-entropy loss to measure prediction divergence and evaluates performance with top-1 accuracy and AUC-ROC. For regression tasks, it applies Mean Squared Error (MSE) for accuracy and calculates Spearman and Pearson correlation coefficients to assess relationships.

These benchmarks [52, 27] offer a thorough evaluation of GFMs. However, all these benchmarks overlook the safety aspects of the GFMs. Recently, the safety of large scientific foundation models has become a prominent focus in research [45, 74]. As a groundbreaking approach to incorporating adversarial attacks into genomic data analysis, FIMBA [74] leverages publicly available genomic datasets, such as The Cancer Genome Atlas (TCGA) and COVID-19 single-cell RNA sequencing data, to assess the robustness of AI models against adversarial feature importance attacks. In the TCGA dataset, the classification task aims to determine whether a sample is malignant, while in the COVID-19 dataset, the objective is to identify whether a patient is diagnosed with the disease. As part of this evaluation, FIMBA uses Accuracy as the primary performance metric to measure the classification capability. To assess the quality and stealth of the adversarial attacks, they employ the Structural Similarity Index Measure (SSIM). SSIM quantifies the structural similarity between

the original and adversarially attacked data, with higher values indicating attacks that are more undetectable and preserve the data's original structure.

C.2 Adversarial Attack

Adversarial attacks can be broadly classified into untargeted, targeted, and universal attacks. Untargeted attacks [87, 51, 79, 38, 56, 59] aim to cause any misprediction by modifying the input in the direction of the loss gradient, maximizing overall loss. In contrast, targeted attacks [90, 44, 14, 6, 77, 5] guide the model's output toward a specific attacker-defined class using the loss gradient directed at the target class. Universal attacks [74, 86, 89, 68, 61, 37, 60] generate perturbations applicable to any input from a given class, causing mispredictions universally.

The Fast Gradient Sign Method (FGSM) [51] and Projected Gradient Descent (PGD) [57] are two prominent techniques for generating adversarial examples in machine learning, particularly for deep neural networks [72]. FGSM generates adversarial samples by applying a single-step perturbation in the direction of the gradient of the loss function, scaled to a predefined magnitude, making it computationally efficient. However, PGD improves robustness by iteratively applying small gradient-based perturbations while ensuring that adversarial examples remain within a specified norm constraint, leading to more effective attacks.

A variety of adversarial attack and defense strategies have recently been proposed, specifically tailored for natural language processing (NLP) tasks [26]. These techniques can be categorized into character-level, word-level, and sentence-level adversarial attacks. Character-level adversarial attacks involve perturbing individual characters in text to mislead machine learning models while preserving readability. For example, DeepWordBug [24] modifies specific characters based on importance scores to maximize the model's misclassification while minimizing changes to the text. Similarly, TextBugger [40] generates adversarial examples by replacing, inserting, or removing characters, focusing on semantic preservation and evading detection by defense mechanisms. Word-level adversarial attacks focus on perturbing entire words rather than individual characters. These attacks can be broadly classified into three categories: gradient-based, importance-based, and replacement-based methods. Gradient-based methods, such as FGSM [51], utilize gradients to identify vulnerable words and modify them to maximize the model's loss. Importance-based methods, exemplified by TextFooler [33], rank words based on their contribution to the model's prediction and replace them with semantically similar alternatives to alter the output. Replacement-based methods, like BERT-Attack [42], leverage pre-trained language models to generate context-aware substitutions, ensuring the adversarial examples maintain fluency and semantic coherence. Sentence-level adversarial attacks involve generating adversarial examples by modifying entire sentences to mislead the model while maintaining grammaticality and semantic relevance. AdvGen [7] generates adversarial sentences by leveraging reinforcement learning to iteratively modify sentence structures and word choices, ensuring the adversarial examples remain coherent and natural while effectively deceiving the target model.

Adversarial attacks have also been explored in genomic models to assess their robustness and identify vulnerabilities in sequence-based predictions. FIMBA [74] presents a black-box, model-agnostic attack and analysis framework designed for widely used machine learning models in genomics. FIMBA targets genomic models by perturbing key features identified through SHAP values, which measure the importance of each feature to the model's decision. By selecting the most impactful features and modifying them using interpolation between the original and target vectors, FIMBA generates minimally altered adversarial examples that effectively deceive the model. The attack avoids gradient reliance, functioning as a black-box method, and focuses on modifying as few features as possible to ensure both high efficacy and low detectability.

C.3 Defense Methods

To improve the robustness of GFMs, various defense strategies [36, 54, 2, 97, 10, 39, 66] are proposed, including adversarial training, defensive distillation, adversarial sample detection, and regularization, purification, and certified robustness. Among these, adversarial training [2, 97, 91, 57] is the most effective, enhancing model resilience by injecting adversarial examples during training. Among these methods, Madry et al. [56] propose a method to inject bounded perturbations into word embeddings and minimize worst-case loss, almost halving BERT-Attack and TextFooler

success rates without degrading clean accuracy. FreeLB [97] merges several PGD steps into one forward-backward pass and accumulates gradients, cutting training cost; FreeLB++ [47] enlarges the radius and steps for further robustness gains at no extra accuracy loss. Other lightweight variants such as SMART[32], TAVAT [41], and R3F [1] approximate the inner maximization with uncertainty- or noise-based regularization, reaching performance close to FreeLB++ at a fraction of the compute. The frequency-aware randomization framework ADFAR [2] incorporates anomaly-detection signals and word-frequency constraints directly into the training loop, unifying adversarial sample detection ideas with adversarial training to further weaken substitution-based attacks without extra overhead. Defensive distillation [18, 66] trains a student model on softened outputs from a teacher model to smooth decision boundaries, though its efficacy against strong adversarial attacks remains debated. However, Carlini and Wagner [4] demonstrate that defensive distillation is ineffective against adaptive adversarial attacks, as carefully crafted inputs can still bypass the smoothed decision boundaries and fool the model. Adversarial sample detection [10, 76, 39, 20] focuses on identifying malicious inputs rather than improving model robustness. MAFD [34] combines perplexity, word frequency, and masking-probability features for robust anomaly scoring; ONION [69] leverages language-model perplexity to prune high-risk tokens; Sharpness-based detectors [93] add infinitesimal noise and flag samples exhibiting steep loss increases. Deployed alongside adversarial training, these detectors offer real-time protection against unseen or cross-domain attacks. Regularization, purification and certified Robustness reduce perturbation sensitivity by modifying the loss or sanitizing inputs. Flooding-X [50] maintains a loss floor to guide the model toward flatter regions; adversarial label smoothing [83] and temperature scaling [81] curb over-confidence; masked-language-model purification [43] masks and reconstructs suspicious tokens to cleanse perturbations. Interval bound propagation (IBP) [31] and randomized smoothing schemes such as SAFER [84] and RanMASK [88] provide formal guarantees against word substitutions or masking budgets.

D Ethical Considerations

Prior to making this work public, we share our adversarial attack artefacts and our results with leading GFMs teams, as shown in Appendix G. Secondly, we open-source the code and data used in our experiments to promote transparency. Also, we carefully consider the ethical impact of our work and list the two impacts: (1) The adversarial sample released in the **GenoAdv** dataset and experiments can provide incorrect classification for existing GFMs. (2) Adversarial training is an efficiency method to make GFMs more resilient to adversarial attacks.

D.1 Dual-Use and Misuse Risks

We recognize that adversarial attacks on genomic foundation models (GFMs), particularly those applied to clinical diagnostics and gene pathogenicity prediction, raise significant dual-use and misuse concerns. While our intention is to improve the safety and robustness of GFMs, we acknowledge that, if misused, the techniques developed in this work could be repurposed to evade genomic screening, manipulate diagnostic predictions, or interfere with treatment decision-making.

The adversarial samples included in the **GenoAdv** dataset are designed to reveal vulnerabilities in current models by targeting biologically meaningful regions. These vulnerabilities highlight the urgency for robust defensive strategies. However, we also recognize that releasing such resources without caution could present opportunities for malicious use.

To mitigate these risks, we take the following steps. First, we have contacted several leading GFM development teams to disclose our findings and foster collaboration on model hardening. Second, although we open-sourced our code and data to promote reproducibility, we now include a usage statement specifying that the tools and dataset are intended strictly for non-commercial research purposes. Use in clinical or diagnostic applications, or for purposes that could impact public health, is explicitly discouraged.

We urge future researchers to approach this line of work with similar responsibility. Any use of GenoAdv or our attack pipeline should be guided by ethical principles that prioritize model reliability, biosecurity, and societal benefit. Our overarching goal is not to facilitate harm, but to proactively identify and close security gaps in genomic models before they can be exploited in real-world settings.

E Reproducibility

In this section, we provide a discussion on the reproducibility of our experiments, including the details of the datasets used, the training and evaluation protocols, and the hyperparameters employed in our experiments.

E.1 Source of Randomness.

To ensure reproducibility, we run all experiments using three different random seeds. We observe that the results are highly stable, with the benchmark introducing only minor variations—showing a variance of at most 2%.

E.2 Implementation.

To ensure reproducibility, we implement the adversarial attack and defense methods based on their official GitHub repositories, as shown below:

- **BertAttack:** https://github.com/LinyangLee/BERT-Attack
- **TextFooler:** https://github.com/jind11/TextFooler
- **PGD:** https://github.com/MadryLab/robustness
- **FIMBA:** https://github.com/HeorhiiS/fimba-attack
- **ADFAR:** https://github.com/LilyNLP/ADFAR
- **FreeLB:** https://github.com/zhuchen03/FreeLB

E.3 Hyperparameter.

We present the hyperparameters used in the benchmark for each model. We use **AdamW** [53] as the optimizer. Fine-tuning and adversarial training are performed uniformly across all models and datasets for 4 epochs, using a batch size of 64 and a maximum sequence length of 256. We use the AdamW optimizer with a learning rate of $3e^{-5}$, gradient accumulation steps of 1, and a warmup ratio of 0.05. The maximum sequence length and batch size used for each adversarial attack and defense method are summarized in Table 5. These settings are chosen to balance computational efficiency and attack effectiveness across different methods.

Table 5: Hyperparameter settings for each attack method.

Hyperparameter	BertAttack	TextFooler	PGD	FIMBA	ADFAR	FreeLB
Max Sequence Length	128	256	256	128	128	256
Batch Size	32	128	16	32	2	32

For **BertAttack**, we configure the attack with $\mathtt{k} = 48$ and set the prediction score threshold to 0, using DNABERT-2 as the reference masked language model. In **ADFAR's** frequency-aware randomization process, we set the frequency threshold $f_{\text{thres}} = 200$, the number of samples $n_s = 20$, and the number of features $n_f = 10$. For **FreeLB**, the hyperparameters used in our experiments include an adversarial learning rate of 0.1, adversarial magnitude of 0.6, two adversarial steps, a base learning rate of $1e^{-5}$, gradient accumulation steps set to 1, and a weight decay of $1e^{-2}$.

F Additional GenoArmory demonstration

We provide two installation options for GenoArmory and two usage methods: via command line and Python code.

Example of Installation of GenoArmory

```
# Install with pip
pip install genoarmory

# Install with source code
git clone https://github.com/MAGICS-LAB/GenoArmory.git
conda create -n genoarmory pip=3.9
pip install .
```

Example of Python Usage of GenoArmory

```
# Initialize model
from GenoArmory import GenoArmory
import json
# You need to initialize GenoArmory with a model and tokenizer.
gen = GenoArmory(model=None, tokenizer=None)
params_file = 'xxx/scripts/PGD/pgd_dnabert.json'

# Visulization
gen.visualization(
    folder_path='xxx/BERT-Attack/results/meta/test',
    output_pdf_path='xxx/BERT-Attack/results/meta/test'
)

# Attack
if params_file:
  try:
      with open(params_file, "r") as f:
          kwargs = json.load(f)
  except json.JSONDecodeError as e:
      raise ValueError(f"Invalid JSON in params file")
  except FileNotFoundError:
      raise FileNotFoundError(f"Params file not found.")

gen.attack(
    attack_method='pgd',
    model_path='magicslabnu/GERM',
    **kwargs
)
```

Example of Commend Line Usage of GenoArmory

```
# Attack
python GenoArmory.py
--model_path magicslabnu/GERM attack
--method pgd --params_file xxx/scripts/PGD/pgd_dnabert.json


# Defense
python GenoArmory.py
--model_path magicslabnu/GERM defense
--method at --params_file xxx/scripts/AT/at_pgd_dnabert.json

# Visualization
python GenoArmory.py
--model_path magicslabnu/GERM visualize
--folder_path xxx/BERT-Attack/results/meta/test
--save_path xxx/BERT-Attack/results/meta/test/frequency.pdf


# Read MetaData
python GenoArmory.py
--model_path magicslabnu/GERM read
--type attack --method TextFooler --model_name dnabert
```

G Disclosure

We share our disclosure with the authors of DNABERT-2, NT, HyenaDNA, and GenomeOcean to inform them of our findings and benchmark. Also, we highlight the potential impact on their models in our disclosure.

Example of Disclosure Letter

Dear DNABERT/DNABERT-2/DNABERT-S team,

We hope this message finds you well. We are reaching out to share the preliminary results and artifacts from our recent study on adversarial attacks targeting DNA-based Genomic Foundation Models (GFMs), which we plan to release publicly as part of a unified benchmarking framework.

Given your leading role in the development of GFMs, we believe it is essential to disclose our findings to you in advance. Our results demonstrate that carefully crafted adversarial sequences can induce incorrect classifications across multiple GFM architectures. We also find that adversarial training remains a promising defense strategy for enhancing model robustness.

To support responsible disclosure, we are providing:

1. A summary of key findings and model vulnerabilities
2. The adversarial sample set and evaluation scripts
3. A description of our ethical considerations and intended safeguards

We welcome your feedback on potential risks, mitigation strategies, and collaborative opportunities to ensure this research contributes constructively to the GFM community. Please let us know if you would like early access to the materials or would prefer to schedule a meeting to discuss further.

Best regards,
GenoArmory Author

H Disclosure of LLM Usage

We utilize Cursor to assist in writing repetitive bash automation scripts and employ GPT-4o to refine the paper's language for conciseness and precision.

I Experiment Setting

I.1 Computational Resource

We perform all experiments using 4 NVIDIA H100 GPUs with 80GB of memory and a 24-core Intel(R) Xeon(R) Gold 6338 CPU operating at 2.00 GHz.

I.2 Metrics of Experiments

In our experiments, we use two core metrics to evaluate the effectiveness of adversarial attacks and the robustness of defense strategies: **Attack Success Rate (ASR)** and **Defense Success Rate (DSR)**.

Attack Success Rate (ASR) is defined as the relative drop in accuracy caused by the adversarial attack. Formally, let A_{clean} be the model accuracy on clean inputs and A_{adv} be the accuracy on adversarial inputs, then:

$$\text{ASR} = \frac{A_{\text{clean}} - A_{\text{adv}}}{A_{\text{clean}}} \times 100\%. \tag{I.1}$$

Defense Success Rate (DSR) measures the robustness gain achieved by applying a defense mechanism. Let A_{def} be the accuracy of the defended model on adversarial inputs, then:

$$\text{DSR} = (1 - \frac{A_{\text{def}} - A_{\text{adv}}}{A_{\text{def}}}) \times 100\%. \tag{I.2}$$

These metrics allow us to quantitatively assess both the impact of adversarial attacks and the degree to which defenses can mitigate that impact.

I.3 Implementation

For DNABERT-2, we use the 117-million-parameter version of the model[2]. For NT, we use the 2.5-billion-parameter version of the model[3]. For NT2, we use the 100-million-parameter version of the model[4]. For HyenaDNA, we use the 4.07-million-parameter version of the model[5]. All four models represent state-of-the-art approaches for genome sequence classification tasks, consistently achieving high performance across various datasets. GenomeOcean [96], on the other hand, is a transformer-based model designed explicitly for genome sequence generation tasks, demonstrating superior performance compared to existing models, such as Evo [63]. We use the 100-million-parameter version of the model[6]. For our experiments, we fine-tuned all of these models using their official checkpoints on the datasets employed in this study.

I.4 Downstream Tasks Across Different Models

We examine the downstream tasks of several genomic foundation models (GFMs), including DNABERT-2 [94], HyenaDNA [64], GenomeOcean [96], and Nucleotide Transformer [12]. As summarized in Table 6, these models primarily focus on classification tasks. In contrast, our analysis of the GenBench datasets [52] reveals the inclusion of regression tasks, offering a more comprehensive evaluation framework.

Table 6: **Comparison of Models (Benchmarks) and Their Tasks.**

Model	Tasks	Classification-Only
DNABERT-2	GUE (28 Classification tasks)	Yes
Nucleotide Transformer	Nucleotide Transformer Benchmark (18 Classification tasks)	Yes
HyenaDNA	GenBench (Classification-Only) + Nucleotide Transformer Benchmark	Yes
GenomeOcean	Classification + Generation (5 GUE Classification tasks)	No
GenBench	Classification + Regression (e.g., Drosophila Enhancer Activity Prediction)	No

J Additional Numerical Experiments

J.1 All results in Adversarial Attack

This section provides a comprehensive evaluation of multiple adversarial attacks across different GFM models. We compare BertAttack, TextFooler, FIMBA, and PGD on a range of biolGenomeOceanical prediction tasks, including epigenetic marks prediction, promoter detection, and transcription factor prediction in both human and mouse datasets. The evaluated GFM models include DNABERT-2, NT, NT2, HyenaDNA, and GenomeOcean.

[2] zhihan1996/DNABERT-2-117M
[3] InstaDeepAI/nucleotide-transformer-2.5b-multi-species
[4] InstaDeepAI/nucleotide-transformer-v2-100m-multi-species
[5] LongSafari/hyenadna-small-32k-seqlen-hf
[6] pGenomeOcean/GenomeOcean-100M

Table 7: **Performance Comparison of Adversarial Attacks on DNABERT-2.** This table shows the performance of all adversarial attacks on the DNABNERT-2 model. All results are evaluated using the Attack Success Rate (ASR) metric. The best result is highlighted in bold, while the second-best result is underlined.

	Epigenetic Marks Prediction					
Attack	H3	H3K14ac	H3K36me3	H3K4me1	H3K4me2	H3K4me3
BertAttack	**91.20**	99.70	99.80	**95.10**	**99.20**	99.30
TextFooler	90.40	**99.90**	**99.90**	86.50	**99.20**	**100.00**
FIMBA	43.70	51.90	24.00	41.30	26.90	41.70
PGD	41.30	33.30	35.50	35.90	38.40	31.80

	Epigenetic Marks Prediction				Promoter Detection (300bp)		
Attack	H3K79me3	H3K9ac	H4	H4ac	all	notata	tata
BertAttack	97.50	**98.00**	**96.60**	**100.00**	**83.70**	**92.70**	96.50
TextFooler	**99.40**	96.20	96.00	94.20	71.80	28.30	**97.00**
FIMBA	24.40	43.80	36.60	50.60	58.30	14.90	87.10
PGD	41.40	39.30	36.20	46.10	45.60	43.50	42.90

	Transcription Factor Prediction (Human)					Core Promoter Detection		
Attack	tf0	tf1	tf2	tf3	tf4	all	notata	tata
BertAttack	96.80	97.60	99.80	90.20	97.40	**99.20**	**99.30**	**98.90**
TextFooler	96.40	**98.00**	**99.40**	**91.30**	**98.80**	97.40	97.10	92.00
FIMBA	50.00	34.10	55.60	25.40	45.30	44.00	32.10	28.20
PGD	36.60	32.30	35.60	34.80	41.00	35.10	34.10	35.80

	Transcription Factor Prediction (Mouse)				
Attack	0	1	2	3	4
BertAttack	93.40	**96.40**	96.20	90.90	**96.90**
TextFooler	**94.20**	94.50	**97.20**	**92.40**	94.20
FIMBA	46.40	3.10	43.30	46.40	39.50
PGD	43.50	38.80	35.10	45.40	36.00

Table 8: **Performance Comparison of Adversarial Attacks on HyenaDNA.** This table shows the performance of all adversarial attacks on the HyenaDNA model. All results are evaluated using the Attack Success Rate (ASR) metric. The best result is highlighted in bold, while the second-best result is underlined.

	Epigenetic Marks Prediction					
Attack	H3	H3K14ac	H3K36me3	H3K4me1	H3K4me2	H3K4me3
BertAttack	**100.00**	**100.00**	**100.00**	**99.06**	**100.00**	**100.00**
TextFooler	**100.00**	**100.00**	**100.00**	92.70	**100.00**	91.14
FIMBA	46.27	3.17	3.51	16.13	14.81	8.20
PGD	10.70	6.70	91.14	5.11	90.68	4.45

	Epigenetic Marks Prediction				Promoter Detection (300bp)		
Attack	H3K79me3	H3K9ac	H4	H4ac	all	notata	tata
BertAttack	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	97.06	**100.00**
TextFooler	35.79	41.68	**100.00**	99.19	46.49	**99.19**	92.85
FIMBA	25.86	38.10	18.18	35.48	48.68	31.17	41.67
PGD	7.04	12.23	22.12	2.58	25.13	92.41	93.72

	Transcription Factor Prediction (Human)					Core Promoter Detection		
Attack	tf0	tf1	tf2	tf3	tf4	all	notata	tata
BertAttack	**100.00**	99.88	**100.00**	**100.00**	98.81	**100.00**	**100.00**	**100.00**
TextFooler	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
FIMBA	38.16	35.71	31.94	26.39	48.86	34.15	32.14	33.33
PGD	90.42	92.86	93.24	90.70	96.65	24.47	12.25	93.59

	Transcription Factor Prediction (Mouse)				
Attack	0	1	2	3	4
BertAttack	**100.00**	99.97	**100.00**	**100.00**	98.79
TextFooler	0.74	**100.00**	**100.00**	**100.00**	**100.00**
FIMBA	40.79	40.22	36.59	32.84	26.67
PGD	0.00	4.35	2.65	90.99	90.18

Table 9: **Performance Comparison of Adversarial Attacks on NT.** This table shows the performance of all adversarial attacks on the Nucleotide Transformer (NT) model. All results are evaluated using the Attack Success Rate (ASR) metric. The best result is highlighted in bold, while the second-best result is underlined.

	Epigenetic Marks Prediction					
Attack	H3	H3K14ac	H3K36me3	H3K4me1	H3K4me2	H3K4me3
BertAttack	**99.92**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
TextFooler	66.23	**100.00**	92.29	97.32	**100.00**	**100.00**
FIMBA	55.13	42.65	25.00	22.06	39.06	31.67
PGD	38.53	38.45	39.11	36.16	36.93	25.25

	Epigenetic Marks Prediction				Promoter Detection (300bp)		
Attack	H3K79me3	H3K9ac	H4	H4ac	all	notata	tata
BertAttack	**100.00**	**100.00**	**99.24**	**100.00**	**100.00**	**100.00**	**100.00**
TextFooler	**100.00**	**100.00**	90.70	89.24	99.19	**100.00**	91.20
FIMBA	30.77	36.36	58.89	32.20	57.45	44.90	46.51
PGD	40.91	20.45	38.24	39.11	36.14	35.47	36.70

	Transcription Factor Prediction (Human)					Core Promoter Detection		
Attack	tf0	tf1	tf2	tf3	tf4	all	notata	tata
BertAttack	**100.00**	**100.00**	99.72	**100.00**	**100.00**	99.76	99.55	99.27
TextFooler	**100.00**	**100.00**	**100.00**	**100.00**	95.39	**100.00**	**100.00**	**100.00**
FIMBA	37.33	41.98	30.99	20.90	43.04	33.80	35.23	42.86
PGD	46.85	48.61	34.57	39.56	53.13	38.24	39.04	57.08

	Transcription Factor Prediction (Mouse)				
Attack	0	1	2	3	4
BertAttack	**100.00**	**99.66**	99.46	**100.00**	**100.00**
TextFooler	**100.00**	92.47	**100.00**	**100.00**	**100.00**
FIMBA	35.71	51.06	39.02	16.36	28.13
PGD	26.10	41.97	37.61	45.96	23.91

Table 10: **Performance Comparison of Adversarial Attacks on NT2.** This table shows the performance of all adversarial attacks on the Nucleotide Transformer 2 (NT2) model. All results are evaluated using the Attack Success Rate (ASR) metric. The best result is highlighted in bold, while the second-best result is underlined.

	Epigenetic Marks Prediction					
Attack	H3	H3K14ac	H3K36me3	H3K4me1	H3K4me2	H3K4me3
BertAttack	98.42	99.62	99.91	99.66	**100.00**	**100.00**
TextFooler	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
FIMBA	27.38	22.08	34.48	30.26	23.53	39.71
PGD	43.55	35.86	16.13	11.19	38.99	11.95

	Epigenetic Marks Prediction				Promoter Detection (300bp)		
Attack	H3K79me3	H3K9ac	H4	H4ac	all	notata	tata
BertAttack	**100.00**	99.53	99.45	**100.00**	99.70	**95.35**	99.47
TextFooler	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	88.59	**100.00**
FIMBA	6.02	62.03	23.08	25.61	59.60	9.09	51.58
PGD	34.78	38.82	32.60	38.35	35.34	32.95	18.03

	Transcription Factor Prediction (Human)					Core Promoter Detection		
Attack	tf0	tf1	tf2	tf3	tf4	all	notata	tata
BertAttack	**100.00**	**100.00**	**100.00**	**99.83**	**100.00**	99.63	99.31	**99.64**
TextFooler	**100.00**	**100.00**	88.84	99.80	**100.00**	**99.81**	**100.00**	40.23
FIMBA	44.71	28.95	37.18	33.75	50.55	45.35	34.48	44.79
PGD	50.82	65.69	45.11	36.52	63.40	11.81	37.73	37.70

	Transcription Factor Prediction (Mouse)				
Attack	0	1	2	3	4
BertAttack	**100.00**	99.59	**99.49**	**100.00**	**100.00**
TextFooler	99.78	**99.82**	95.74	**100.00**	97.84
FIMBA	50.00	42.71	40.70	38.89	42.50
PGD	38.69	40.22	15.00	41.88	21.56

Table 11: **Performance Comparison of Adversarial Attacks on GenomeOcean.** This table shows the performance of all adversarial attacks on the GenomeOcean model. All results are evaluated using the Attack Success Rate (ASR) metric. The best result is highlighted in bold, while the second-best result is underlined.

	Epigenetic Marks Prediction					
Attack	H3	H3K14ac	H3K36me3	H3K4me1	H3K4me2	H3K4me3
BertAttack	**100.00**	99.60	99.97	**100.00**	99.95	99.97
TextFooler	99.78	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
FIMBA	45.88	36.14	24.10	49.35	53.73	51.95
PGD	47.74	42.41	41.11	48.82	38.28	45.57

	Epigenetic Marks Prediction				Promoter Detection (300bp)		
Attack	H3K79me3	H3K9ac	H4	H4ac	all	notata	tata
BertAttack	98.75	**100.00**	**98.18**	98.51	99.65	**100.00**	97.71
TextFooler	**100.00**	**100.00**	88.89	**100.00**	**99.87**	**100.00**	**100.00**
FIMBA	43.37	21.52	35.16	68.67	59.78	36.36	28.57
PGD	44.12	48.49	43.45	18.72	53.34	41.15	35.22

	Transcription Factor Prediction (Human)					Core Promoter Detection		
Attack	tf0	tf1	tf2	tf3	tf4	all	notata	tata
BertAttack	**100.00**	**100.00**	**99.89**	99.60	99.94	99.83	99.91	99.81
TextFooler	**100.00**	**100.00**	99.88	**99.85**	**100.00**	**100.00**	**100.00**	**100.00**
FIMBA	46.91	31.65	49.37	39.39	45.88	42.68	31.33	38.96
PGD	22.98	22.98	23.95	33.33	22.06	41.39	32.15	39.66

	Transcription Factor Prediction (Mouse)				
Attack	0	1	2	3	4
BertAttack	**100.00**	99.83	98.95	98.83	**100.00**
TextFooler	**100.00**	**99.89**	**100.00**	**99.90**	**100.00**
FIMBA	1.16	53.68	34.83	57.65	39.47
PGD	43.36	23.68	24.94	32.90	38.91

Table 12: **Performance Comparison of Adversarial Defense on DNABERT-2.** This table shows the performance of all adversarial defense on the DNABERT-2 model. All results are evaluated using the Defense Success Rate (DSR) metric. The best result is highlighted in bold, while the second-best result is underlined.

		Epigenetic Marks Prediction					
Attack	Defense	H3	H3K14ac	H3K36me3	H3K4me1	H3K4me2	H3K4me3
PGD	FreeLB	56.17	65.68	66.22	63.10	72.38	63.92
	ADFAR	**64.32**	63.55	65.51	62.01	74.57	**64.58**
	AT	54.87	**77.97**	**69.08**	**72.55**	**82.38**	61.01
BertAttack	FreeLB	5.10	0.00	1.16	0.00	1.19	10.00
	ADFAR	**100.00**	0.00	**10.10**	0.00	2.08	**94.23**
	AT	4.76	0.00	0.00	0.00	**2.86**	0.00
TextFooler	FreeLB	33.88	0.11	0.00	0.00	0.00	0.00
	ADFAR	**42.28**	0.00	0.00	0.00	0.00	**0.22**
	AT	41.25	**0.12**	**0.12**	0.00	**1.88**	0.00

		Epigenetic Marks Prediction				Promoter Detection (300bp)		
Attack	Defense	H3K79me3	H3K9ac	H4	H4ac	all	notata	tata
PGD	FreeLB	61.47	**63.44**	60.84	**67.58**	55.93	56.01	58.74
	ADFAR	62.08	55.82	65.56	62.38	**70.01**	**65.59**	**64.26**
	AT	**62.91**	60.92	**73.12**	59.48	63.67	51.98	49.74
BertAttack	FreeLB	0.00	1.08	6.19	0.00	0.00	1.00	**9.28**
	ADFAR	0.00	**8.42**	0.00	**25.00**	**4.08**	**100.00**	7.69
	AT	**4.55**	4.29	**15.62**	0.00	2.04	19.59	8.75
TextFooler	FreeLB	0.00	0.00	34.68	0.00	0.00	3.04	73.16
	ADFAR	0.00	0.00	**76.39**	**4.74**	**8.42**	**100.00**	**88.83**
	AT	**1.28**	**5.57**	38.16	0.00	0.00	28.97	75.63

		Transcription Factor Prediction (Human)					Core Promoter Detection		
Attack	Defense	tf0	tf1	tf2	tf3	tf4	all	notata	tata
PGD	FreeLB	**66.17**	**72.23**	73.21	**66.79**	**65.54**	73.30	69.31	**64.18**
	ADFAR	64.78	64.38	56.85	56.18	61.97	60.61	67.32	59.56
	AT	64.44	64.76	**77.58**	59.93	57.08	**74.31**	**76.35**	62.18
BertAttack	FreeLB	**10.20**	0.00	10.00	**2.15**	2.27	0.00	0.00	0.00
	ADFAR	0.00	0.00	0.00	0.00	**100.00**	**27.08**	**7.07**	0.00
	AT	0.00	0.00	**10.34**	0.00	0.00	1.20	1.14	**1.10**
TextFooler	FreeLB	0.22	0.00	0.00	0.34	0.71	0.00	1.01	72.85
	ADFAR	0.00	0.00	**6.29**	**100.00**	2.41	**26.29**	**1.61**	**97.97**
	AT	**0.98**	0.00	0.24	0.13	**3.17**	0.00	0.66	75.18

		Transcription Factor Prediction (Mouse)				
Attack	Defense	0	1	2	3	4
PGD	FreeLB	57.93	70.40	56.17	57.29	**61.82**
	ADFAR	**69.44**	64.73	60.40	**62.41**	61.54
	AT	55.61	**73.15**	**73.22**	53.08	56.45
BertAttack	FreeLB	20.62	4.12	**9.00**	**17.35**	2.20
	ADFAR	**44.44**	**27.27**	0.00	10.42	**100.00**
	AT	5.49	10.20	6.82	5.71	1.10
TextFooler	FreeLB	65.90	0.00	85.89	89.98	16.28
	ADFAR	67.49	**17.54**	**91.92**	**96.23**	**26.15**
	AT	**68.2**	6.18	87.45	92.45	17.54

Table 13: **Performance Comparison of Adversarial Defense on GenomeOcean.** This table shows the performance of all adversarial defense on the GenomeOcean model. All results are evaluated using the Defense Success Rate (DSR) metric. The best result is highlighted in bold, while the second-best result is underlined.

		Epigenetic Marks Prediction					
Attack	Defense	H3	H3K14ac	H3K36me3	H3K4me1	H3K4me2	H3K4me3
PGD	FreeLB	**58.51**	50.75	52.96	55.52	58.13	**56.48**
	ADFAR	54.75	**66.59**	49.43	**68.20**	**69.17**	50.24
	AT	57.40	55.78	**59.35**	49.87	64.69	52.15
BertAttack	FreeLB	**2.04**	**8.60**	**3.19**	0.00	0.00	0.00
	ADFAR	0.00	0.00	0.00	0.00	0.00	0.00
	AT	0.22	4.45	0.13	**0.04**	**0.22**	0.00
TextFooler	FreeLB	0.00	0.00	0.00	0.00	0.00	0.00
	ADFAR	0.00	0.00	0.00	0.00	0.00	0.00
	AT	**33.75**	0.00	0.00	0.00	0.00	0.00

		Epigenetic Marks Prediction				Promoter Detection (300bp)		
Attack	Defense	H3K79me3	H3K9ac	H4	H4ac	all	notata	tata
PGD	FreeLB	**57.31**	55.04	56.79	**93.99**	45.78	52.47	63.83
	ADFAR	51.14	46.38	**61.72**	86.29	52.74	51.28	64.24
	AT	56.04	**55.60**	56.85	92.58	**53.46**	**65.77**	**66.48**
BertAttack	FreeLB	**6.12**	**22.99**	**24.24**	**1.05**	0.00	0.00	0.00
	ADFAR	0.00	0.00	0.00	0.00	0.00	**3.77**	0.00
	AT	1.05	1.69	1.31	0.75	0.00	0.04	0.00
TextFooler	FreeLB	0.00	0.00	35.25	0.00	0.00	0.00	73.8
	ADFAR	0.00	0.00	0.51	0.00	0.00	**100.00**	**100.00**
	AT	0.00	0.00	**37.13**	0.00	0.00	0.00	74.10

		Transcription Factor Prediction (Human)					Core Promoter Detection		
Attack	Defense	tf0	tf1	tf2	tf3	tf4	all	notata	tata
PGD	FreeLB	**96.51**	91.09	91.18	67.74	91.79	**70.92**	66.86	57.31
	ADFAR	92.09	**97.83**	93.73	67.07	**96.94**	57.38	58.62	55.30
	AT	91.54	93.95	**94.37**	**68.14**	96.53	60.82	**69.29**	**61.32**
BertAttack	FreeLB	0.00	0.00	0.00	0.00	**1.00**	**2.15**	0.00	**1.01**
	ADFAR	0.00	0.00	0.00	0.00	0.00	1.85	0.00	0.00
	AT	0.00	0.00	0.00	0.00	0.00	0.76	**0.80**	0.18
TextFooler	FreeLB	0.00	0.00	0.00	0.00	0.00	0.00	0.00	73.13
	ADFAR	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	0.00	0.00	0.10
	AT	0.00	0.52	0.00	0.00	0.42	0.00	0.00	**74.49**

		Transcription Factor Prediction (Mouse)				
Attack	Defense	0	1	2	3	4
PGD	FreeLB	57.25	**73.37**	68.87	67.39	57.16
	ADFAR	55.60	69.74	**69.72**	**68.53**	57.96
	AT	**58.48**	70.22	48.47	61.68	**58.82**
BertAttack	FreeLB	0.00	1.05	2.00	1.00	0.00
	ADFAR	0.00	**25.00**	0.00	0.00	0.00
	AT	0.00	0.00	**2.02**	**2.00**	0.00
TextFooler	FreeLB	64.44	0.00	85.73	89.57	28.76
	ADFAR	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
	AT	65.98	1.65	85.47	90.03	17.63

Table 14: **Performance Comparison of Adversarial Defense on NT.** This table shows the performance of all adversarial defense on the Nucleotide Transformer (NT) model. All results are evaluated using the Defense Success Rate (DSR) metric. The best result is highlighted in bold, while the second-best result is underlined.

		Epigenetic Marks Prediction					
Attack	Defense	H3	H3K14ac	H3K36me3	H3K4me1	H3K4me2	H3K4me3
PGD	FreeLB	87.79	84.45	80.44	85.20	84.35	74.08
	ADFAR	54.65	53.07	50.08	57.23	54.72	57.95
	AT	**92.35**	**86.74**	**82.02**	**86.80**	**87.54**	**75.02**
BertAttack	FreeLB	**7.14**	0.00	0.00	**1.18**	0.00	0.00
	ADFAR	2.04	0.00	0.00	0.00	**13.56**	0.00
	AT	0.22	0.00	0.00	0.00	0.00	0.00
TextFooler	FreeLB	25.69	23.10	10.30	12.40	20.00	**9.54**
	ADFAR	0.00	**100.00**	**62.70**	**12.90**	9.35	7.33
	AT	**47.68**	24.97	12.31	9.39	**47.97**	7.99

		Epigenetic Marks Prediction				Promoter Detection (300bp)		
Attack	Defense	H3K79me3	H3K9ac	H4	H4ac	all	notata	tata
PGD	FreeLB	**85.64**	**83.36**	**89.65**	**84.87**	93.93	95.41	**99.34**
	ADFAR	53.70	61.02	59.09	59.92	52.09	51.25	57.63
	AT	84.74	81.22	82.81	81.39	**94.26**	**96.97**	90.45
BertAttack	FreeLB	0.00	0.00	2.06	0.00	0.00	0.00	2.02
	ADFAR	**6.52**	0.00	**2.17**	0.00	0.00	**43.75**	**11.76**
	AT	0.00	0.00	2.04	0.00	0.00	1.02	0.00
TextFooler	FreeLB	**22.17**	41.03	62.86	35.14	35.79	31.25	85.07
	ADFAR	2.55	**100.00**	**72.48**	**42.77**	**49.20**	**69.14**	**91.32**
	AT	13.34	24.61	53.74	23.82	35.97	34.56	82.09

		Transcription Factor Prediction (Human)					Core Promoter Detection		
Attack	Defense	tf0	tf1	tf2	tf3	tf4	all	notata	tata
PGD	FreeLB	57.58	55.28	72.30	**82.95**	48.23	**85.07**	89.47	39.77
	ADFAR	**96.49**	**97.26**	**92.94**	59.67	**96.92**	54.34	56.28	**95.70**
	AT	84.21	59.95	66.17	62.06	64.31	81.73	**91.93**	38.15
BertAttack	FreeLB	0.00	0.00	0.00	0.00	0.00	1.04	**1.06**	**1.02**
	ADFAR	0.00	0.00	0.00	**5.66**	0.00	0.00	0.00	0.00
	AT	0.00	0.00	0.00	0.00	0.00	**1.15**	0.00	**1.02**
TextFooler	FreeLB	36.03	34.17	32.44	28.15	38.83	44.54	47.10	89.26
	ADFAR	**100.00**	**60.02**	**75.00**	**99.58**	**89.74**	**64.10**	**100.00**	**100.00**
	AT	43.85	41.76	28.65	44.43	37.16	34.18	35.66	86.49

		Transcription Factor Prediction (Mouse)				
Attack	Defense	0	1	2	3	4
PGD	FreeLB	74.60	98.03	86.32	**70.60**	75.08
	ADFAR	56.57	55.57	53.62	52.30	59.28
	AT	**76.37**	**99.44**	**99.46**	34.72	**75.64**
BertAttack	FreeLB	0.00	0.00	0.00	**2.02**	0.00
	ADFAR	0.00	**41.07**	**2.13**	0.00	0.00
	AT	0.00	0.13	0.00	0.00	0.00
TextFooler	FreeLB	75.28	58.13	92.57	93.98	31.72
	ADFAR	**85.24**	**83.82**	**97.60**	**100.00**	**69.05**
	AT	72.34	56.08	89.80	94.44	31.63

Table 15: **Performance Comparison of Adversarial Defense on NT2.** This table shows the performance of all adversarial defense on the Nucleotide Transformer-2 (NT2) model. All results are evaluated using the Defense Success Rate (DSR) metric. The best result is highlighted in bold, while the second-best result is underlined.

		Epigenetic Marks Prediction					
Attack	Defense	H3	H3K14ac	H3K36me3	H3K4me1	H3K4me2	H3K4me3
PGD	FreeLB	89.10	80.99	79.18	84.75	**76.23**	76.21
	ADFAR	86.57	73.38	77.85	77.95	55.52	67.38
	AT	**97.61**	**82.31**	**83.20**	**86.88**	75.67	**77.66**
BertAttack	FreeLB	**2.02**	0.00	0.00	0.00	0.00	0.00
	ADFAR	0.00	**5.97**	**1.67**	0.00	0.00	0.00
	AT	0.00	0.00	0.00	0.00	0.00	0.00
TextFooler	FreeLB	33.23	0.00	0.00	0.00	0.00	0.00
	ADFAR	**49.57**	0.00	0.00	0.00	0.00	0.00
	AT	35.70	0.00	0.00	0.00	0.00	0.00

		Epigenetic Marks Prediction				Promoter Detection (300bp)		
Attack	Defense	H3K79me3	H3K9ac	H4	H4ac	all	notata	tata
PGD	FreeLB	**89.83**	84.55	**99.44**	79.34	**94.93**	**91.57**	**94.00**
	ADFAR	89.28	73.77	74.60	73.34	61.27	57.61	70.18
	AT	89.43	**86.33**	96.76	**83.78**	93.74	90.42	85.16
BertAttack	FreeLB	0.00	0.00	3.12	0.00	0.00	0.00	**1.00**
	ADFAR	**18.18**	0.00	**4.26**	0.00	0.00	**18.75**	0.00
	AT	0.00	0.00	1.02	0.00	0.00	0.00	0.00
TextFooler	FreeLB	0.00	**0.11**	35.29	0.00	**0.71**	0.00	73.73
	ADFAR	0.00	0.00	**72.82**	0.00	0.00	**0.71**	**76.05**
	AT	0.00	0.00	35.22	0.00	0.00	0.00	74.33

		Transcription Factor Prediction (Human)					Core Promoter Detection		
Attack	Defense	tf0	tf1	tf2	tf3	tf4	all	notata	tata
PGD	FreeLB	**92.86**	**94.01**	**82.76**	84.25	**97.22**	**91.26**	**91.33**	**99.82**
	ADFAR	73.62	68.87	73.46	71.17	75.97	73.68	78.40	62.35
	AT	61.98	68.87	61.98	**87.24**	94.12	88.43	76.66	55.54
BertAttack	FreeLB	0.00	0.00	0.00	**2.22**	0.00	0.00	0.00	0.00
	ADFAR	**51.06**	**60.38**	0.00	0.00	**2.04**	0.00	0.00	0.00
	AT	0.00	4.00	**1.00**	0.00	0.00	0.00	0.00	**1.00**
TextFooler	FreeLB	0.00	0.00	0.00	0.00	0.00	0.00	0.00	72.76
	ADFAR	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**85.48**
	AT	0.00	0.00	**0.15**	**0.12**	0.00	0.00	0.00	77.81

		Transcription Factor Prediction (Mouse)				
Attack	Defense	0	1	2	3	4
PGD	FreeLB	**88.71**	**99.19**	97.29	81.65	**81.44**
	ADFAR	77.22	74.06	**99.56**	66.95	61.05
	AT	74.61	97.07	**99.56**	**86.09**	51.29
BertAttack	FreeLB	0.00	**4.04**	**2.00**	**4.08**	0.00
	ADFAR	**1.92**	0.00	0.00	0.00	**16.67**
	AT	0.00	4.00	1.00	0.00	0.00
TextFooler	FreeLB	63.98	0.00	85.96	89.66	16.67
	ADFAR	**77.00**	0.00	86.34	**94.90**	**29.07**
	AT	67.30	0.20	**86.69**	92.44	22.71

Table 16: **Performance Comparison of Adversarial Defense on HyenaDNA.** This table shows the performance of all adversarial defense on the HyenaDNA model. All results are evaluated using the Defense Success Rate (DSR) metric. The best result is highlighted in bold, while the second-best result is underlined.

		Epigenetic Marks Prediction					
Attack	Defense	H3	H3K14ac	H3K36me3	H3K4me1	H3K4me2	H3K4me3
PGD	FreeLB	76.72	70.87	98.19	91.86	96.22	85.29
	ADFAR	**88.44**	74.31	85.63	**94.41**	**98.83**	84.20
	AT	**88.44**	**84.26**	**99.36**	86.77	91.96	**87.48**
BertAttack	FreeLB	0.00	0.00	0.00	0.00	0.00	0.00
	ADFAR	0.00	0.00	0.00	0.00	0.00	0.00
	AT	0.00	0.00	0.00	0.00	0.00	0.00
TextFooler	FreeLB	**100.00**	98.08	71.00	**75.21**	53.82	**100.00**
	ADFAR	**100.00**	**99.77**	30.70	50.62	29.01	97.75
	AT	**100.00**	84.18	**95.87**	50.68	**64.87**	80.81

		Epigenetic Marks Prediction				Promoter Detection (300bp)		
Attack	Defense	H3K79me3	H3K9ac	H4	H4ac	all	notata	tata
PGD	FreeLB	95.32	90.09	62.33	85.31	56.04	**94.81**	**97.27**
	ADFAR	93.53	**98.33**	60.58	**95.96**	83.52	40.20	89.77
	AT	**96.32**	93.99	**63.34**	85.31	**98.47**	49.07	76.80
BertAttack	FreeLB	0.00	0.00	0.00	0.00	0.00	**16.33**	0.00
	ADFAR	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	AT	0.00	0.00	0.00	0.00	0.00	10.00	0.00
TextFooler	FreeLB	20.42	17.94	90.50	3.28	76.54	**100.00**	93.46
	ADFAR	63.24	15.85	88.98	81.68	65.48	92.86	89.93
	AT	**99.64**	**45.01**	**92.80**	**85.59**	**100.00**	27.44	**93.97**

		Transcription Factor Prediction (Human)					Core Promoter Detection		
Attack	Defense	tf0	tf1	tf2	tf3	tf4	all	notata	tata
PGD	FreeLB	**87.44**	87.44	88.44	87.44	**88.44**	**98.47**	85.47	**96.26**
	ADFAR	83.42	**99.50**	76.38	**95.48**	87.44	68.94	**98.61**	90.77
	AT	**87.44**	87.44	**91.46**	87.44	79.40	96.10	**98.61**	83.30
BertAttack	FreeLB	2.13	0.00	2.04	0.00	0.00	**6.82**	0.00	**1.92**
	ADFAR	0.00	0.00	0.00	0.00	0.00	1.85	0.00	0.00
	AT	**5.98**	0.00	**3.72**	0.00	0.00	0.00	0.00	0.00
TextFooler	FreeLB	23.33	19.42	95.63	**100.00**	14.08	66.42	99.38	94.70
	ADFAR	**100.00**	**100.00**	89.68	87.00	**100.00**	93.89	**100.00**	**100.00**
	AT	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**

		Transcription Factor Prediction (Mouse)				
Attack	Defense	0	1	2	3	4
PGD	FreeLB	**94.47**	**98.59**	**75.38**	**83.76**	89.81
	ADFAR	85.43	87.08	62.81	65.99	87.68
	AT	**94.47**	97.23	62.81	65.99	**94.09**
BertAttack	FreeLB	0.00	0.00	0.00	0.00	0.00
	ADFAR	**37.04**	0.00	0.00	0.00	0.00
	AT	1.23	0.00	0.00	0.00	0.00
TextFooler	FreeLB	**100.00**	19.69	**100.00**	94.94	**80.78**
	ADFAR	**100.00**	**89.64**	**100.00**	**100.00**	35.34
	AT	**100.00**	37.31	**100.00**	**100.00**	31.02

Table 17: **Performance Comparison of Adversarial Attack on Quantization Model.** This table reports the Attack Success Rate (ASR) of two adversarial attacks (TextFooler and BERTAttack) on quantized versions (Vanilla and Softmax$_1$) of DNABERT-2 and Nucleotide Transformer (NT) under W8A8 (8-bit weights and activations) quantization. All results are evaluated using the Attack Success Rate (ASR) metric.

			Epigenetic Marks Prediction					
Attack	Model	Quant_Method	H3	H3K14ac	H3K36me3	H3K4me1	H3K4me2	H3K4me3
TextFooler	DNABERT2	Vanilla	**0.19**	**9.76**	**24.12**	**5.52**	25.53	**12.24**
		Softmax$_1$	0.00	3.82	15.67	2.03	**31.90**	4.14
	NT1	Vanilla	70.49	**79.37**	**77.74**	**77.04**	**70.49**	**87.14**
		Softmax$_1$	**73.96**	73.65	77.53	70.89	70.33	86.21
BertAttack	DNABERT2	Vanilla	**62.50**	26.09	**100.00**	61.54	81.25	**100.00**
		Softmax$_1$	**62.50**	**100.00**	16.00	**100.00**	**93.75**	60.00
	NT1	Vanilla	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
		Softmax$_1$	92.31	**100.00**	**100.00**	**100.00**	**100.00**	99.60

			Epigenetic Marks Prediction				Promoter Detection (300bp)		
Attack	Model	Quant_Method	H3K79me3	H3K9ac	H4	H4ac	all	notata	tata
TextFooler	DNABERT2	Vanilla	**4.30**	0.00	**11.48**	1.30	27.58	17.05	**30.29**
		Softmax$_1$	3.96	0.00	4.19	**1.48**	**28.21**	**22.44**	29.55
	NT1	Vanilla	**71.49**	**73.37**	**56.52**	**72.17**	59.54	54.59	**58.15**
		Softmax$_1$	68.89	67.25	55.12	71.90	**68.42**	**63.40**	**58.15**
BertAttack	DNABERT2	Vanilla	**100.00**	**100.00**	**57.14**	**99.78**	**98.08**	**96.43**	72.56
		Softmax$_1$	84.62	87.50	0.00	96.15	66.11	70.00	**100.00**
	NT1	Vanilla	**100.00**	**100.00**	91.67	**100.00**	98.25	93.75	**100.00**
		Softmax$_1$	**100.00**	**100.00**	**99.27**	**100.00**	**100.00**	**97.83**	**100.00**

			Transcription Factor Prediction (Human)					Core Promoter Detection		
Attack	Model	Quant_Method	tf0	tf1	tf2	tf3	tf4	all	notata	tata
TextFooler	DNABERT2	Vanilla	1.17	0.00	**14.07**	38.34	0.20	**63.88**	**67.90**	**61.33**
		Softmax$_1$	**13.45**	**5.61**	11.49	**38.67**	**4.48**	62.36	61.12	48.87
	NT1	Vanilla	57.41	51.93	67.28	74.05	53.26	**66.18**	63.73	42.81
		Softmax$_1$	**69.22**	**65.50**	**71.97**	**77.68**	**69.39**	59.52	**68.14**	**49.06**
BertAttack	DNABERT2	Vanilla	0.00	**11.11**	**63.64**	**100.00**	16.67	**97.83**	**64.29**	89.02
		Softmax$_1$	**2.91**	2.91	26.58	80.00	**32.47**	96.71	36.79	**98.55**
	NT1	Vanilla	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
		Softmax$_1$	**100.00**	96.43	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	99.60

Table 18: **Performance of Adversarial Attacks on HyenaDNA Trained with the GenoAdv Dataset.** This table compares the performance of HyenDNA trained with adversarial examples from the GenoAdv dataset. Three attack methods (BERTAttack, TextFooler, and PGD) are used to evaluate the models, with results reported in terms of Attack Success Rate (ASR). The best result is highlighted in bold, while the second-best result is underlined.

	Epigenetic Marks Prediction					
Attack	H3	H3K14ac	H3K36me3	H3K4me1	H3K4me2	H3K4me3
TextFooler	1.01	5.41	<u>83.24</u>	<u>3.18</u>	<u>17.86</u>	<u>62.82</u>
PGD	<u>12.83</u>	<u>19.29</u>	17.20	2.85	4.73	6.13
BERT_Attack	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**

	Epigenetic Marks Prediction				Promoter Detection (300bp)		
Attack	H3K79me3	H3K9ac	H4	H4ac	all	notata	tata
TextFooler	<u>26.27</u>	<u>45.20</u>	<u>33.53</u>	<u>94.53</u>	<u>44.20</u>	<u>26.00</u>	1.05
PGD	12.56	16.90	20.16	7.71	21.13	10.06	<u>20.27</u>
BERT_Attack	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**

	Transcription Factor Prediction (Human)					Core Promoter Detection		
Attack	tf0	tf1	tf2	tf3	tf4	all	notata	tata
TextFooler	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
PGD	<u>3.70</u>	**40.00**	<u>19.15</u>	<u>22.22</u>	<u>19.15</u>	<u>3.11</u>	<u>13.83</u>	<u>9.81</u>
BERT_Attack	**70.37**	<u>15.00</u>	**100.00**	**100.00**	**100.00**	**83.02**	**100.00**	**95.74**

	Transcription Factor Prediction (Mouse)				
Attack	0	1	2	3	4
TextFooler	0.00	0.00	0.00	0.00	<u>23.94</u>
PGD	<u>44.44</u>	<u>7.06</u>	<u>17.45</u>	<u>15.79</u>	14.90
BERT_Attack	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**

Table 19: **Performance of Adversarial Attacks on GenomeOcean Trained with the GenoAdv Dataset.** This table compares the performance of GenomeOcean trained with adversarial examples from the GenoAdv dataset. Three attack methods (BERTAttack, TextFooler, and PGD) are used to evaluate the models, with results reported in terms of Attack Success Rate (ASR). The best result is highlighted in bold, while the second-best result is underlined.

	Epigenetic Marks Prediction					
Attack	H3	H3K14ac	H3K36me3	H3K4me1	H3K4me2	H3K4me3
TextFooler	<u>62.66</u>	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
PGD	34.44	35.87	24.51	40.00	39.43	1.36
BERT_Attack	**100.00**	<u>98.56</u>	<u>97.65</u>	**100.00**	**100.00**	**100.00**

	Epigenetic Marks Prediction				Promoter Detection (300bp)		
Attack	H3K79me3	H3K9ac	H4	H4ac	all	notata	tata
TextFooler	**100.00**	**100.00**	<u>63.89</u>	**100.00**	**100.00**	**100.00**	22.65
PGD	39.52	36.69	26.34	34.64	33.45	34.76	<u>30.91</u>
BERT_Attack	<u>95.70</u>	**100.00**	**97.94**	<u>98.77</u>	**100.00**	<u>96.45</u>	**100.00**

	Transcription Factor Prediction (Human)					Core Promoter Detection		
Attack	tf0	tf1	tf2	tf3	tf4	all	notata	tata
TextFooler	**100.00**	**100.00**	**100.00**	**100.00**	<u>99.89</u>	<u>98.32</u>	**100.00**	22.71
PGD	34.18	12.68	35.80	19.15	35.65	44.22	40.89	<u>39.07</u>
BERT_Attack	<u>98.12</u>	**100.00**	**100.00**	**100.00**	**100.00**	**98.84**	**100.00**	**100.00**

	Transcription Factor Prediction (Mouse)				
Attack	0	1	2	3	4
TextFooler	24.73	<u>96.33</u>	13.58	8.88	<u>80.71</u>
PGD	<u>35.06</u>	30.33	<u>34.42</u>	<u>26.60</u>	25.45
BERT_Attack	**100.00**	**100.00**	**98.96**	**100.00**	**100.00**

Table 20: **Performance of Adversarial Attacks on DNABERT-2 Trained with the GenoAdv Dataset.** This table compares the performance of DNABERT-2 trained with adversarial examples from the GenoAdv dataset. Three attack methods (BERTAttack, TextFooler, and PGD) are used to evaluate the models, with results reported in terms of Attack Success Rate (ASR). The best result is highlighted in bold, while the second-best result is underlined.

	Epigenetic Marks Prediction					
Attack	H3	H3K14ac	H3K36me3	H3K4me1	H3K4me2	H3K4me3
TextFooler	61.83	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
PGD	39.53	24.67	34.53	36.71	35.61	34.79
BERT_Attack	**87.67**	85.36	**100.00**	88.63	88.13	**100.00**

	Epigenetic Marks Prediction				Promoter Detection (300bp)		
Attack	H3K79me3	H3K9ac	H4	H4ac	all	notata	tata
TextFooler	**99.88**	69.87	61.00	**100.00**	56.26	**100.00**	24.27
PGD	41.24	29.06	26.35	37.59	38.23	45.11	44.93
BERT_Attack	88.90	**100.00**	**87.10**	**100.00**	**100.00**	88.99	**87.56**

	Transcription Factor Prediction (Human)					Core Promoter Detection		
Attack	tf0	tf1	tf2	tf3	tf4	all	notata	tata
TextFooler	**100.00**	99.87	**100.00**	**100.00**	**99.21**	**100.00**	**100.00**	23.39
PGD	30.12	25.33	24.39	2.22	28.09	36.36	22.71	36.89
BERT_Attack	95.60	**100.00**	**100.00**	97.78	98.88	**100.00**	98.80	**100.00**

	Transcription Factor Prediction (Mouse)				
Attack	0	1	2	3	4
TextFooler	28.54	**98.28**	12.77	6.49	81.43
PGD	35.81	30.25	9.64	13.00	34.63
BERT_Attack	**100.00**	87.94	**87.59**	**96.61**	**100.00**

Table 21: **Performance of Adversarial Attacks on NT Trained with the GenoAdv Dataset.** This table compares the performance of Nucleotide Transformers (NT) trained with adversarial examples from the GenoAdv dataset. Three attack methods (BERTAttack, TextFooler, and PGD) are used to evaluate the models, with results reported in terms of Attack Success Rate (ASR). The best result is highlighted in bold, while the second-best result is underlined.

	Epigenetic Marks Prediction					
Attack	H3	H3K14ac	H3K36me3	H3K4me1	H3K4me2	H3K4me3
TextFooler	56.41	70.39	77.72	85.08	77.87	80.64
PGD	28.57	23.43	21.88	29.53	21.67	22.90
BERT_Attack	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**

	Epigenetic Marks Prediction				Promoter Detection (300bp)		
Attack	H3K79me3	H3K9ac	H4	H4ac	all	notata	tata
TextFooler	79.42	69.67	52.19	66.39	46.25	64.64	21.50
PGD	17.64	26.87	7.49	19.89	19.39	7.97	7.83
BERT_Attack	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**

	Transcription Factor Prediction (Human)					Core Promoter Detection		
Attack	tf0	tf1	tf2	tf3	tf4	all	notata	tata
TextFooler	58.31	61.81	46.13	60.44	67.96	44.69	67.92	13.82
PGD	28.57	24.15	21.57	25.48	10.11	23.01	25.96	13.01
BERT_Attack	**100.00**	**85.37**	**100.00**	**97.85**	**98.88**	**100.00**	**100.00**	**100.00**

	Transcription Factor Prediction (Mouse)				
Attack	0	1	2	3	4
TextFooler	24.55	76.23	10.08	8.26	66.19
PGD	25.00	21.96	10.71	26.81	26.46
BERT_Attack	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**

Table 22: **Performance of Adversarial Attacks on NT2 Trained with the GenoAdv Dataset.** This table compares the performance of Nucleotide Transformers-2 (NT2) trained with adversarial examples from the GenoAdv dataset. Three attack methods (BERTAttack, TextFooler, and PGD) are used to evaluate the models, with results reported in terms of Attack Success Rate (ASR). The best result is highlighted in bold, while the second-best result is underlined.

	Epigenetic Marks Prediction					
Attack	H3	H3K14ac	H3K36me3	H3K4me1	H3K4me2	H3K4me3
TextFooler	<u>65.28</u>	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
PGD	29.13	23.43	21.88	29.53	31.75	22.90
BERT_Attack	**100.00**	**100.00**	<u>99.84</u>	**100.00**	<u>95.67</u>	**100.00**

	Epigenetic Marks Prediction				Promoter Detection (300bp)		
Attack	H3K79me3	H3K9ac	H4	H4ac	all	notata	tata
TextFooler	**100.00**	**100.00**	<u>63.67</u>	**100.00**	<u>53.67</u>	**100.00**	<u>24.35</u>
PGD	24.51	26.87	28.29	22.67	29.39	2.19	13.01
BERT_Attack	**100.00**	**100.00**	**91.56**	**100.00**	**100.00**	**100.00**	**100.00**

	Transcription Factor Prediction (Human)					Core Promoter Detection		
Attack	tf0	tf1	tf2	tf3	tf4	all	notata	tata
TextFooler	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	24.50
PGD	22.17	21.76	26.96	23.33	26.32	45.80	28.48	<u>28.69</u>
BERT_Attack	<u>99.81</u>	**100.00**	<u>98.91</u>	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**

	Transcription Factor Prediction (Mouse)				
Attack	0	1	2	3	4
TextFooler	<u>31.09</u>	**100.00**	13.31	8.88	<u>80.71</u>
PGD	9.09	28.69	<u>13.56</u>	<u>26.81</u>	28.02
BERT_Attack	**100.00**	<u>98.99</u>	**100.00**	**100.00**	**100.00**

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 abstract and introduction clearly state the main claims made in the paper.

   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: We discuss the Limitations in Section 5.

   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]

Justification: The paper does not include theoretical results.

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: We provide all the details necessary to reproduce the main experimental results in the paper. The details of the experiments are included in both the main paper and the supplemental material.

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: [Yes]

Justification: We open-source the code and data with detailed instructions to reproduce the main experimental results. We provide the code in Appendix A.

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: We introduce the experimental setting in main paper and provide additional details in the supplemental material.

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: We report the variance in all tables and observed stable results with less than 2% variance.

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: We introduce the compute resources in Appendix I.1.

   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: We have reviewed the NeurIPS Code of Ethics and confirm that the research conducted in the paper conforms to it. We discuss the Ethical Consideration in Appendix D.

   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: We discuss the broader impacts in Appendix B.

    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: [No]

    Justification: We provide a dataset containing adversarial attack samples on GFMs as part of GenoAdv. However, due to dataset access policy requirements in Neuips, it is publicly available at this time. Following the NeurIPS review period, we may enable access upon approval to prevent potential misuse.

    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 assets used in the paper are properly attributed to their original creators, and we explicitly mention and respect the licenses and terms of use in Appendix I.3.

    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: We publicly release all models and datasets along with detailed documentation in Appendix A.

    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]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    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]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    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: We disclose LLM usage in Appendix H.

    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.