# Guarding the Life Code: Preserving Membership Privacy in Genomic Foundation Models

**Xinyu Zhao**[1], **Jinhao Duan**[1], **Zhen Tan**[2], **Tianlong Chen**[*1]

[1]UNC Chapel Hill [2]Stevens Institute of Technology

## Abstract

Genomic Foundation Models (GFMs) have become a promising paradigm for decoding DNA sequences, yet their privacy risks are critically underexplored. When fine-tuned on human cohorts, GFMs can unintentionally memorize sensitive genomic signatures that enable membership inference attacks (MIAs) and expose individuals to long-lasting privacy harms. However, existing defenses often fall short in genomics: globally applied mechanisms can substantially degrade motif-sensitive utility, and model-agnostic post-processing provides limited insight into *which* genomic regions drive leakage. Through a systematic privacy audit, we find that membership leakage in GFMs is highly non-uniform, concentrating on a small subset of high-risk samples and sparse token spans corresponding to biologically meaningful patterns. Motivated by this finding, we propose `GenoGuard`, a model- and sample-adaptive defense that shifts from blanket protection to targeted, region-aware mitigation. `GenoGuard` (i) localizes privacy-leaking subsequences via gradient-based token attribution and performs token-level risk reduction through selective gradient routing, (ii) applies sample-level risk reduction with risk-adaptive label smoothing guided by a reference-calibrated loss gap score, and (iii) provides region-level attribution maps for privacy auditing and biological interpretation. Experiments across three representative GFMs (Mistral-DNA, Nucleotide Transformer, HyenaDNA) demonstrate that `GenoGuard` consistently improves privacy against multiple MIA families while preserving strong fine-tuning performance and interpretability. `GenoGuard` is an empirical defense complementary to differential privacy: stacked on DP-SGD, it further reduces leakage under the same formal $(\varepsilon, \delta)$ budget. Codes are provided in `https://github.com/UNITES-Lab/GenoGuard.git`.

## 1 Introduction

Genomic Foundation Models (GFMs) have emerged as a new paradigm in computational biology by adapting the architecture and scaling laws of transformer-based Large Language Models (LLMs) to decipher DNA sequences [23, 49, 10, 31, 37]. By pretraining on large-scale genomic corpora, GFMs provide transferable representations that enable a wide range of downstream tasks, such as regulatory element prediction, epigenetic mark prediction, variant effect analysis, and functional annotation [49, 33, 15]. However, the rapid adoption and deployment of GFMs also introduces pressing privacy challenges, since training data often comes from human cohorts with sensitive phenotypes, rare traits, or identifiable ancestry information [29, 13].

A central concern is *unintended memorization*: GFMs trained on sensitive cohorts may inadvertently encode individual-specific genomic patterns, such as rare variants, disease-associated motifs, or ancestry markers, into their parameters [6, 7]. Once memorized, such information can be exposed by *membership inference attacks* (MIAs) [39, 44, 19], which reveal whether a specific individual's data was included in training and can thereby disclose sensitive health or familial information. This risk is

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

particularly acute in genomics, as genomic data is immutable and inherently identifying, leading to lifelong consequences for affected individuals and their relatives [29, 17].

Despite this urgency, privacy risks specific to GFMs remain underexplored. While MIAs have been extensively studied in computer vision and natural language processing [38, 36, 40, 6], genomic sequences exhibit unique statistical properties—small alphabet size (four nucleotides), structured regulatory motifs, and long-range dependencies that may fundamentally alter attack surfaces and memorization patterns compared to natural language or image data [31, 37, 15]. Moreover, prior genomics-focused privacy studies have largely centered on classical statistical methods rather than modern foundation models [19, 18, 8, 47]. The threat is already practical. Biobanks such as All of Us and UK Biobank [41, 5] follow a model-to-data design: raw genomes stay inside a secure enclave, but models trained there may be exported after an output review that screens for raw-record and small-count disclosure, not for membership signal stored in model parameters. MIAs already succeed against clinical language models [21] and medical diagnostic models, where some individual patients are identified with near-perfect success [25]. As GFMs increasingly become shared resources via model hubs and APIs, understanding and mitigating their membership leakage becomes necessary. This leads us to ask:

***(Q)** How vulnerable are GFMs to membership inference attacks? And how can we design defenses that mitigate privacy leakage while preserving utility and interpretability?*

Existing defenses fall short in the genomic context for two reasons. First, the dominant formal defense, differentially private SGD (DP-SGD) [1], supplies a worst-case $(\varepsilon, \delta)$ guarantee by uniformly clipping and noising every per-example gradient. This blanket protection erases precisely the subtle, biologically meaningful signals that downstream tasks rely on, such as transcription-factor binding sites or promoter activity prediction, where labels hinge on short motifs or single-nucleotide substitutions [8, 47]. Second, most other MIA defenses are model-agnostic and operate at a global granularity: they regularize overall training dynamics or post-process output confidences to obfuscate membership signals [24, 20, 22]. Such defenses treat the model as a black box and cannot localize which specific subsequences drive the leakage, even though, as we show in Section 4, leakage in fine-tuned GFMs is sharply concentrated on a small fraction of samples and on sparse, high-information spans such as rare variants or short regulatory motifs. We therefore design an empirical, targeted defense that is *complementary* to DP-SGD rather than a replacement: it reduces the leakage that already passes through any practical $(\varepsilon, \delta)$ budget, while preserving utility on motif-sensitive tasks. Where a worst-case bound is required, DP-SGD remains necessary, and our defense stacks on it.

To address question **(Q)**, we conduct a systematic privacy audit of GFMs (Section 4). Our analysis yields a key insight that guides our defense design: membership leakage in GFMs is highly non-uniform. It concentrates on a small subset of high-risk samples and, within each sample, on sparse token spans corresponding to specific genomic patterns. Motivated by these findings, we propose `GenoGuard`, a model and sample adaptive privacy-preserving framework for GFMs that addresses both limitations simultaneously. `GenoGuard` first localizes privacy-leaking subsequences to provide region-level interpretability, and then performs targeted risk reduction to avoid blanket penalties, specifically through a two-fold strategy: (1) **Token-Level Risk Reduction**: we identify memorization-prone token spans via gradient-based token attribution and selectively downscale their gradients through gradient routing, suppressing leakage sources while preserving benign regions needed for downstream prediction; (2) **Sample-Level Risk Reduction**: we estimate per-sample memorization risk using a reference-calibrated loss gap score and apply risk-adaptive label smoothing to attenuate outlier membership signals with minimal impact on typical samples; and (3) **Region-Level Attribution**: we provide token-span risk maps that highlight regions most responsible for membership evidence, supporting privacy auditing and facilitating biological interpretation. Our main contributions are summarized as follows:

$\star$ We provide the first systematic benchmarking of MIAs and defenses in the GFM setting. Our analysis reveals that privacy leakage is concentrated in rare token patterns and sparse spans that often align with biologically meaningful subsequences (e.g., motif-like patterns).

$\star$ Built on the findings, we propose `GenoGuard`, a defense framework integrating token-level risk-aware gradient scaling with sample-level adaptive smoothing. Unlike existing methods, `GenoGuard` suppresses both loss-based and confidence-based membership signals while retaining strong supervised fine-tuning (SFT) performance on downstream genomic tasks.

⋆ We validate GenoGuard on decoder (Mistral-DNA), encoder (Nucleotide Transformer), and long-convolution (HyenaDNA) backbones across six attack families and four GUE tasks. GenoGuard reduces leakage relative to undefended fine-tuning in every setting, cutting Shadow-attack AUC by up to 28.1 percentage points (81.03% $\rightarrow$ 52.89% on TF-M). Against DP-SGD, it matches overall privacy, attains the lowest Loss-attack AUC in 7 of 8 model–task settings (up to 8.0 points lower), and loses only 1.6 F1 points against DP-SGD's 4.2. Stacked on DP-SGD, it further lowers leakage under the same $(\varepsilon, \delta)$ budget. Its token-level attribution maps localize the regions that drive residual leakage, and these high-risk spans overlap known transcription-factor motifs far more often than low-risk spans ($28.5\%$ vs. $5.2\%$).

## 2 Related Works

**Genomic Foundation Models (GFMs).** Modeling genomic sequences has evolved significantly with the adoption of transformer architectures [12, 4]. Early adaptations, such as the DNABERT series [23, 49], utilize masked language modeling to capture sequence dependencies for biological tasks. Models such as Nucleotide Transformer [10], HyenaDNA [31], GENA-LM [16], and GENERator [43] employ specialized attention mechanisms or recurrent operators to process long sequences efficiently. Later proposed Evo model series [30, 3] bridge the DNA-RNA-protein dogma and enable genome modeling and design across all domains of life, while GenomeOcean [50] scaled to 4 billion parameters for context-aware generation. Recent adaptations like Mistral-DNA [27] demonstrate the viability of fine-tuning general-purpose LLMs for genomic tasks. While these works push model performance, they overlook the privacy implications of memorizing genomic data.

**Membership Inference and Its Defenses.** MIAs evolved from loss and confidence thresholds [44] to shadow models [39, 36] and likelihood-ratio tests [6]; memorization grows with model scale [7], and MIAs succeed on clinical and medical models, with risk concentrated on rare or underrepresented patients [21, 25]. Defenses split by what they guarantee. *Formal* defenses, chiefly DP-SGD [1], bound worst-case membership advantage through uniform per-example clipping and noise, applying the same protection to every sample. *Empirical* defenses, including data augmentation [9], output obfuscation [24, 22], distillation [32], and adversarial regularization [28], reduce measured leakage at lower utility cost without a worst-case bound. GenoGuard belongs to the empirical class: it allocates regularization to the samples and tokens the audit flags, and composes with DP-SGD when a formal bound is required.

**Privacy and Safety in Genomic Foundation Models.** Genomic data is inherently identifiable: a small set of independent SNPs can uniquely re-identify individuals [19, 17], and models trained on stratified populations disproportionately memorize statistical outliers, leaving minority cohorts more exposed to MIA [11, 2]. Privacy threats in genomics have evolved from aggregate statistical attacks to gradient leakage in federated settings [34] and generative risks [2]. Recent work has begun to study safety risks in GFMs, jailbreak attacks that elicit pathogenic sequences [48], adversarial perturbations causing misclassification [26, 46], but membership privacy and defenses tailored to GFMs remain largely unexplored, which is the gap our work closes.

## 3 Preliminaries

### 3.1 Genomic Foundation Models

Genomic foundation models (GFMs) adapt the pre-training paradigm from natural language processing to DNA sequences, learning universal representations from large-scale genomic data that can be transferred to various downstream tasks. Given a DNA sequence $s = (n_1, n_2, \ldots, n_L)$ where $n_i \in \{A, T, C, G\}$, a tokenizer $\mathcal{T}$ maps it into a token sequence $\mathcal{T}(s) = (t_1, t_2, \ldots, t_M)$. Two dominant strategies are k-mer tokenization [10], which uses a fixed-length sliding window, and Byte Pair Encoding (BPE) [49], which iteratively merges frequent nucleotide pairs to form a vocabulary.

**Model Pre-training and Fine-tuning.** A GFM is a Transformer-based encoder or decoder $g_\theta : \mathcal{X} \rightarrow \mathbb{R}^{|\mathcal{V}|}$ that maps a tokenized sequence $\mathcal{T}(s) \in \mathcal{X}$ to per-position logits over the token vocabulary $\mathcal{V}$, with parameters $\theta$. For a downstream classification task with $C$ classes, fine-tuning attaches a linear head and yields a classifier $f_\theta : \mathcal{X} \rightarrow \mathbb{R}^C$ whose softmax $p_\theta(k \,|\, x) = \mathrm{softmax}(f_\theta(x))_k$ defines

the predicted class distribution and $\hat{y} = \arg\max_k f_\theta(x)_k$ is the hard prediction. We use $\ell(f_\theta; x, y)$ for the per-example cross-entropy $-\log p_\theta(y \mid x)$ throughout.

The model is pre-trained on a large corpus $\mathcal{D}_{\text{pretrain}}$ using either Masked Language Modeling (MLM), which predicts masked tokens, or Causal Language Modeling (CLM), which predicts the next token:

$$\mathcal{L}_{\text{MLM}} = \mathbb{E}_{s,\mathcal{M}}\left[-\sum_{i\in\mathcal{M}} \log p_\theta(t_i \mid s_{\setminus\mathcal{M}})\right], \qquad \mathcal{L}_{\text{CLM}} = \mathbb{E}_s\left[-\sum_{i=1}^{M} \log p_\theta(t_i \mid t_{<i})\right], \tag{1}$$

where $\mathcal{M} \subset \{1, \ldots, M\}$ is the masked-position subset for MLM, $s_{\setminus\mathcal{M}}$ is the sequence with masked tokens replaced by a `[MASK]` symbol, and $t_{<i} = (t_1, \ldots, t_{i-1})$ is the causal prefix. After pre-training, the GFM is fine-tuned on a downstream task dataset $\mathcal{D}_{\text{task}} = \{(x_i, y_i)\}_{i=1}^N$ for sequence classification (*e.g.*, promoter prediction), token-level classification (*e.g.*, splice site prediction), or regression.

### 3.2 Threat Model

Given a fine-tuned GFM $f_\theta$ and a data sample $(x, y)$, the goal of a membership inference attack (MIA) is to determine whether $(x, y)$ was part of the training set $\mathcal{D}_{\text{train}}$. This can be formulated as a hypothesis testing problem: the attacker aims to distinguish between the null hypothesis $H_0 : (x, y) \notin \mathcal{D}_{\text{train}}$ and the alternative hypothesis $H_1 : (x, y) \in \mathcal{D}_{\text{train}}$.

**Adversary Access.** We focus on the standard *black-box* setting. The adversary may submit inputs $x$ to the fine-tuned model and observe an output oracle $\mathcal{O}_{f_\theta}(x)$, which returns the logit/probability vector $f_\theta(x) \in \mathbb{R}^C$ (or a softmax thereof), but cannot read internal activations, gradients, or parameters. The membership decision is therefore a function $\mathcal{A}(x, y, \mathcal{O}_{f_\theta}(x)) \to \{0, 1\}$. The per-example loss $\ell(f_\theta; x, y) = -\log p_\theta(y \mid x)$ is computable from $\mathcal{O}_{f_\theta}(x)$ *when the true label $y$ is known to the adversary*, e.g. when the adversary holds a candidate $(x, y)$ pair to test, or when $\mathcal{Y}$ is small enough to enumerate. We assume the adversary additionally has access to non-overlapping samples drawn from the same data distribution, which can be used to train shadow or reference models for likelihood-ratio attacks [39, 6, 36]. We do not assume a white-box adversary that observes parameter updates or per-step gradients; this matches the deployment of GFMs as inference-only APIs, model-hub checkpoints, or models exported from biobank enclaves, and the threat model of every baseline defense.

**Adversary Strategies.** We consider three families that together cover the attack surface relevant to our defense. (1) *Metric-based* attacks compute a membership score directly from model output, e.g., per-example loss, confidence, or full probability vector, and threshold it [44]. (2) *Shadow-model-based* attacks train auxiliary models on the same data distribution and learn a discriminator (Shadow [39, 36], LiRA [6], Quantile regression) over output statistics. (3) *Adaptive* attacks [42] additionally know the defense's structure (e.g., which tokens are routed, the smoothing schedule) and tailor their score accordingly; we treat the standard attacks above as the deployment threat and evaluate adaptive variants separately as a robustness check (Appendix A.9).

**Defender's Capabilities.** The defender controls the fine-tuning pipeline and a held-out validation split disjoint from $\mathcal{D}_{\text{train}}$. They may incorporate privacy-preserving mechanisms during training (loss reweighting, gradient routing, label smoothing, DP-SGD noise) but cannot retrain the pretraining backbone. The defender's objective is to minimize membership leakage under the adversary models above while preserving downstream task utility.

**System Assets and Goal.** The primary asset is the membership privacy of the training data $\mathcal{D}_{\text{train}}$. The attacker builds an inference function $\mathcal{A}(x, y, \mathcal{O}_{f_\theta}(x)) \to \{0, 1\}$ that predicts whether a query sample is a member.

## 4 Problem Statement

### 4.1 Privacy Risks in GFMs

We first run a warm-up study on Mistral-DNA [27] over the GUE benchmark [49], applying both metric-based attacks (Loss/Confidence/Probability) and model-based attacks (Shadow, LiRA, Quantile); attack details are in Appendix A.5. As shown in Figure 2a, every attack exceeds the $50\%$

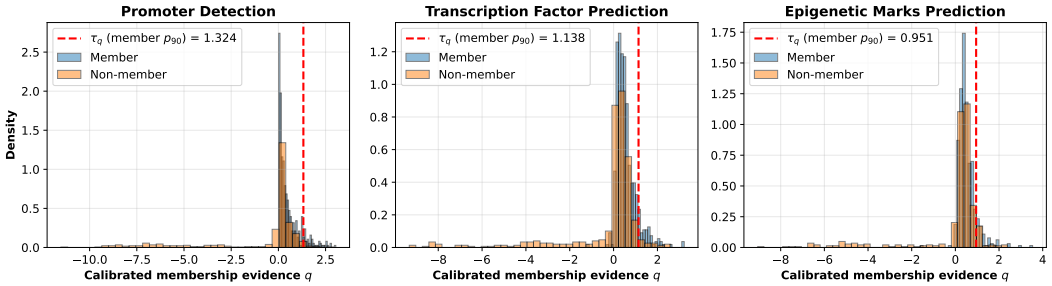
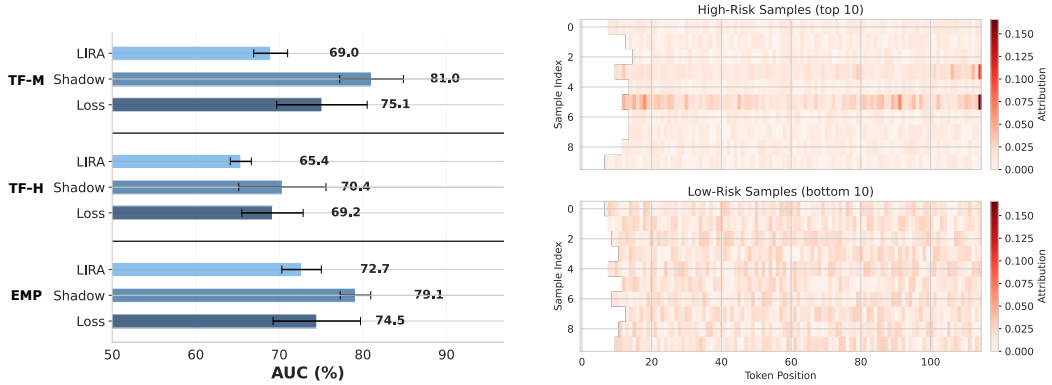



Figure 1: **Membership evidence is heavy-tailed: a small fraction of members carries most of the leakage.** Density of the calibrated membership evidence $q(x, y) = \ell(f_{\theta_{\text{ref}}}; x, y) - \ell(f_{\theta_{\text{tgt}}}; x, y)$ (Eq. 2) for members (blue) vs. non-members (orange) on TF-H, TF-M, and EMP. Notice that the member distributions extend much further into the positive tail than the non-member ones, but only on a thin right tail; the red dashed line marks the member-side $\tau_q = p_{90}$ cutoff that bounds this tail. The defense implication: blanket regularization spends most of its budget on samples that are already indistinguishable from non-members; targeting the right tail can shrink the leakage gap with far less utility cost.




(a) AUC of different attacks on Mistral-DNA across TF-H, TF-M, and EMP.

(b) Token-level gradient attribution heatmaps for high-risk (top, top-10% by $q$) and low-risk (bottom, bottom-10%) member samples on Mistral-DNA / EMP.

Figure 2: **Privacy audit of fine-tuned GFMs.** (a) GFMs leak membership information across attack families. (b) Within member samples, high-risk inputs concentrate attribution on a sparse subset of tokens, while low-risk inputs spread it uniformly.

random-guess line on every task. Shadow is the strongest attack (70.4–81.0%), and the simple Loss attack, which needs no auxiliary models, reaches 69.2–75.1%, above LiRA on every task.

**Observation 1.** Fine-tuned GFMs are vulnerable to MIA, with simple Loss-based attacks reaching up to 75% AUC-ROC. The strong Loss-attack signal motivates a deeper investigation into *where* and *why* memorization occurs.

### 4.2 Dissecting the Root Cause of Privacy Risks in GFM

We locate leakage at two granularities: across samples (Q1) and across tokens within a sample (Q2).

**(Q1) Is leakage uniform across samples?** Although membership is binary, the *strength* of evidence varies. Let $\ell(f_\theta; x, y)$ be the per-example cross-entropy loss; we fine-tune a reference model $f_{\theta_{\text{ref}}}$ on a disjoint split $\mathcal{D}_{\text{ref}}$ ($\mathcal{D}_{\text{ref}} \cap \mathcal{D}_{\text{train}} = \mathcal{D}_{\text{ref}} \cap \mathcal{D}_{\text{test}} = \emptyset$) and define the *calibrated membership evidence*

$$q(x, y) \;=\; \ell(f_{\theta_{\text{ref}}}; x, y) \;-\; \ell\big(f_{\theta_{\text{tgt}}}; x, y\big)\,, \tag{2}$$

so that larger $q(x, y)$ means the target fits $(x, y)$ much better than the reference. Subtracting the reference loss controls for sample difficulty: a hard sample with high target loss can still have $q \approx 0$ if the reference also struggles with it. Figure 1 shows that $q$ has a heavy right tail among members; we mark $\tau_q := p_{90}(\{q(x, y) : (x, y) \in \mathcal{D}_{\text{train}}\})$ as the cutoff for this tail. Long-tailed memorization

has been documented in vision and language [14, 6]; our finding confirms it for GFMs, where tail samples may correspond to rare or minority-cohort sequences.

**Insight 1.** Membership evidence in fine-tuned GFMs is heavy-tailed: a small fraction of samples (right of $\tau_q$) carries most of the leakage. For genomic data, where tail samples may correspond to rare or minority-cohort sequences, this concentration motivates targeted, sample-adaptive mitigation rather than blanket regularization.

**(Q2) Within a high-risk sample, where does leakage live?** We compare token-level attribution patterns between the top-10% and bottom-10% member samples by $q$, using the gradient-based score $s(t_i) = \|\nabla_{\mathbf{e}_i} \mathcal{L}_{\text{cls}} \odot \mathbf{e}_i\|_2$ (defined formally in Eq. (5)). Figure 2b shows that high-risk samples exhibit "hotspot" patterns concentrated on a few positions, while low-risk samples have near-uniform attribution. Quantitatively, on TF-H subtask `tf/0` ($n = 32{,}378$), high-risk samples have significantly higher Gini (0.435 vs. 0.395, $p = 3.8 \times 10^{-7}$, Mann–Whitney U) and lower normalized entropy (0.812 vs. 0.842, $p = 2.1 \times 10^{-7}$), indicating that memorization concentrates on a sparse token subset rather than spreading across the sequence. These hotspots are biologically meaningful: high-risk spans (top-10% by risk score) overlap known transcription-factor motifs from JASPAR [35] at 28.5%, versus 5.2% for low-risk spans and 4.9% for random spans (odds ratio 7.2, $p < 10^{-4}$; Appendix A.10).

**Insight 2:** High-risk samples exhibit concentrated token-level reliance, where the model's prediction hinges on a few positions rather than distributing attention uniformly. This motivates targeted gradient attenuation on high-attribution tokens to prevent overfitting.

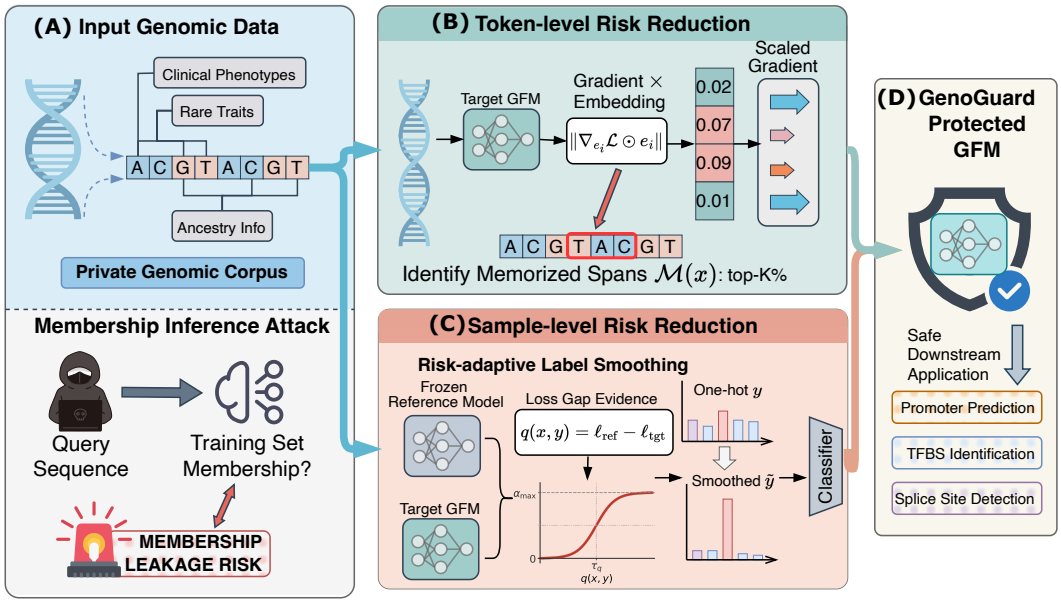



Figure 3: **(A)** A GFM fine-tuned on a private genomic corpus overfits to a small subset of training samples, exposing them to membership inference. **(B) Token-level Risk Reduction:** a gradient-based attribution score from the target GFM identifies the top-$K\%$ tokens of each sequence (red) as the memorization-prone span $\mathcal{M}(x)$; their backward gradient is attenuated by $\gamma$ while the forward representation is unchanged. **(C) Sample-level Risk Reduction:** the loss gap between the frozen reference model and the target GFM is mapped through a sigmoid centered at the member-side cutoff $\tau_q$ into a per-sample smoothing weight $\alpha(x, y)$, which softens the one-hot label only on memorized outliers. **(D)** The protected GFM retains motif-sensitive utility on downstream genomic tasks.

## 5 `GenoGuard`: Adaptive Privacy Defense for Genomic Foundation Models

Our analysis in Section 4 reveals that privacy risk in GFMs is both *sample-heterogeneous* (Insight 1) and *token-localized* (Insight 2). `GenoGuard` mirrors this structure with two cooperating modules

driven by the calibrated evidence $q(x, y)$: **Sample-Level Risk Reduction** converts $q$ into a per-sample label-smoothing weight, and **Token-Level Risk Reduction** uses a gradient-attribution score to attenuate the backward signal on the few tokens the loss is most sensitive to (Figure 3). We assume single-label softmax classification, matching every GUE task we evaluate; multi-label and BCE variants are discussed in Appendix A.3.

### 5.1 Sample-Level Risk Reduction via Adaptive Smoothing

Insight 1 and Observation 1 together motivate a regularizer that targets the loss gap exploited by Loss/Confidence MIAs *only on the high-evidence tail* of the training set. Given the cutoff $\tau_q$ from Section 4, we map each sample to a smoothing weight:

$$\alpha(x, y) \;=\; \texttt{stopgrad}\Big[\; \alpha_{\max} \cdot \sigma\Big(\tfrac{q(x,y)-\tau_q}{t}\Big) \;\Big], \tag{3}$$

where $\sigma(z) = 1/(1 + e^{-z})$ is the logistic sigmoid, $\alpha_{\max} \in (0, 1]$ caps the smoothing strength, and $t > 0$ controls sharpness around $\tau_q$. The sigmoid (rather than a hard threshold) avoids a discontinuity at the cutoff that the optimizer could exploit, and `stopgrad` prevents the loss from back-propagating through $q(x, y)$ into $\theta$. The smoothed target for a $C$-class task is

$$\tilde{y} \;=\; (1 - \alpha(x, y)) \cdot y \;+\; \alpha(x, y) \cdot u, \qquad u_k = \tfrac{1}{C}, \tag{4}$$

where $y$ is the one-hot ground-truth vector and $u$ is uniform over $C$ classes. Outliers ($q \gg \tau_q$) approach $\alpha_{\max}$; typical samples ($q \ll \tau_q$) keep sharp supervision.

### 5.2 Token-Level Risk Reduction via Gradient Routing

Insight 2 shows that memorization within a sample localizes to a sparse set of tokens. We attenuate the optimization signal on those positions while preserving their contribution to the forward representation. For each token $t_i$ with embedding $\mathbf{e}_i \in \mathbb{R}^d$,

$$s(t_i) \;=\; \|\nabla_{\mathbf{e}_i} \mathcal{L}_{\text{cls}} \;\odot\; \mathbf{e}_i\|_2 \,, \tag{5}$$

where $\mathcal{L}_{\text{cls}} = \text{CE}(\text{softmax}(f_\theta(x)), y)$ is the classification cross-entropy on $(x, y)$, $\odot$ is the element-wise product, and $\|\cdot\|_2$ is the Euclidean norm. $s(t_i)$ is the standard gradient $\times$ input attribution and we use it as a proxy for token-level reliance (alternative targets are discussed in Appendix A.3). The high-risk set is taken *per sample* as the top-$K\%$ tokens of each sequence,

$$\mathcal{M}(x) \;=\; \texttt{stopgrad}\big[\; \{\, i \,:\, s(t_i) \;\geq\; \tau_K(x) \,\} \;\big], \qquad \tau_K(x) \;=\; \text{Quantile}_{1-K/100}\big(\{s(t_i)\}_{i=1}^{L}\big). \tag{6}$$

For each position $i \in \mathcal{M}(x)$, we route the input embedding,

$$\mathbf{e}'_i \;=\; \texttt{stopgrad}(\mathbf{e}_i) \;+\; \gamma \cdot \big(\mathbf{e}_i - \texttt{stopgrad}(\mathbf{e}_i)\big), \tag{7}$$

where $\texttt{stopgrad}(\mathbf{v})$ returns $\mathbf{v}$ unchanged in the forward pass and zero in the backward pass, and $\gamma \in [0, 1]$ is the attenuation factor (we use a warm-up schedule $\gamma(e)$, detailed in Appendix A.3); for $i \notin \mathcal{M}(x)$, $\mathbf{e}'_i = \mathbf{e}_i$. Eq. (7) preserves the forward value while scaling the gradient flowing *into* the embedding by $\gamma$. The attenuation is therefore indirect: downstream Transformer parameters depending on $\mathbf{e}'_i$ are not directly scaled, but the embedding moves less per step at high-risk positions, slowing the rate at which the model encodes sample-specific patterns. Because $s(t_i)$ requires a gradient before the routing mask is known, each training step runs a probe pass to compute $\{s(t_i)\}$ and $q(x, y)$, then an actual training pass with $\mathcal{M}(x)$ and $\alpha(x, y)$ frozen via `stopgrad`; full details are in Appendix A.3.

### 5.3 Combining Dual-Level Risk Reduction

`GenoGuard` applies both modules within the same fine-tuning step. Given a mini-batch $\{(x^{(j)}, y^{(j)})\}_{j=1}^{B}$ and the per-sample $\mathcal{M}(x^{(j)})$ and $\alpha(x^{(j)}, y^{(j)})$ from the probe pass, the training pass produces routed embedding sequences $E'^{(j)} = (\mathbf{e}'^{(j)}_1, \ldots, \mathbf{e}'^{(j)}_{L_j})$ via Eq. (7) and smoothed targets $\tilde{y}^{(j)}$ via Eq. (4), and updates $\theta$ on

$$\mathcal{L} \;=\; \tfrac{1}{B} \sum_{j=1}^{B} \text{CE}\Big(\text{softmax}(f_\theta(E'^{(j)})),\; \tilde{y}^{(j)}\Big). \tag{8}$$

The two components act on complementary pathways, optimization and supervision, and can be enabled independently, as we verify in the ablation in Appendix A.7.

Table 1: Performance overview of GenoGuard versus baselines on **Mistral-DNA**. We evaluate attack and defense effectiveness across four downstream tasks: Transcription Factor Prediction (**TF-H, TF-M**), Epigenetic Marks Prediction (**EMP**), and Promoter Detection (**PD**). We apply metric-based attacks (**Loss**, Confidence **Conf.**, Probability **Prob.**) and model-based attacks (**Shadow, LiRA, Quantile**). We report attack AUC-ROC; values closer to $50\%$ indicate stronger privacy protection (random guessing), while values farther from $50\%$ indicate stronger membership signal. Bold marks the defense whose AUC is closest to $50\%$ in each cell.

| Attack | TF-H | | | | TF-M | | | | EMP | | | | PD | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Clean | EMT | DP-SGD | GenoGuard | Clean | EMT | DP-SGD | GenoGuard | Clean | EMT | DP-SGD | GenoGuard | Clean | EMT | DP-SGD | GenoGuard |
| **Loss** | 69.18% | 63.37% | 62.72% | **59.83%** | 75.10% | 67.82% | 64.50% | **56.48%** | 74.48% | 66.89% | 65.80% | **58.79%** | 66.12% | 63.74% | 59.58% | **55.43%** |
| **Conf.** | 62.89% | 60.99% | 60.54% | **57.66%** | 70.32% | 67.32% | **63.14%** | 64.91% | 70.66% | 66.52% | **63.49%** | 64.69% | 62.84% | **54.84%** | 57.90% | 55.41% |
| **Prob.** | 62.83% | 61.76% | **58.20%** | 61.76% | 68.23% | 65.52% | 61.94% | **55.84%** | 69.37% | 68.36% | 65.40% | **58.18%** | 61.14% | 59.50% | **56.73%** | 59.50% |
| **Shadow** | 70.36% | 64.46% | **63.64%** | 64.46% | 81.03% | 71.88% | 55.47% | **52.89%** | 79.11% | 68.40% | 64.04% | **63.95%** | 62.16% | 59.75% | 56.67% | **53.59%** |
| **LIRA** | 65.39% | 64.78% | **64.15%** | 64.85% | 68.96% | 67.88% | 62.33% | **58.41%** | 72.66% | 69.03% | **63.48%** | 64.45% | 60.26% | **56.10%** | 56.31% | **56.10%** |
| **Quantile** | 70.86% | 69.75% | **65.15%** | 65.22% | 62.38% | 59.88% | **55.61%** | 56.45% | 75.62% | 68.09% | **67.99%** | 68.36% | 59.71% | 60.41% | 56.64% | **53.88%** |

Table 2: Downstream utility (F1-score %, higher is better). Best defense in each column is **bold**. GenoGuard stays within 1.3–2.2 points of Clean fine-tuning, whereas DP-SGD loses 2.9–5.2 points.

| | Mistral-DNA | | | | Nucleotide Transformer | | | |
|---|---|---|---|---|---|---|---|---|
| **Defense** | TF-H | TF-M | EMP | PD | TF-H | TF-M | EMP | PD |
| Clean | 81.4 | 68.9 | 85.7 | 90.9 | 84.3 | 80.4 | 70.9 | 96.5 |
| EMT | 78.1 | 66.8 | 82.6 | **89.1** | 81.2 | 77.9 | 67.8 | 94.8 |
| DP-SGD | 76.2 | 64.7 | 81.5 | 86.8 | 79.8 | 76.1 | 66.5 | 93.6 |
| GenoGuard | **79.5** | **67.3** | **84.1** | 88.7 | **82.5** | **78.6** | **69.1** | **95.2** |

# 6 Experiments

Table 3: Performance overview of GenoGuard versus baselines on **Nucleotide Transformer**. We evaluate attack and defense effectiveness across four downstream tasks: Transcription Factor Prediction (**TF-H, TF-M**), Epigenetic Marks Prediction (**EMP**), and Promoter Detection (**PD**). We apply metric-based attacks (**Loss**, Confidence **Conf.**, Probability **Prob.**) and model-based attacks (**Shadow, LiRA, Quantile**). We report attack AUC-ROC; values closer to $50\%$ indicate stronger privacy protection (random guessing), while values farther from $50\%$ indicate stronger membership signal. Bold marks the defense whose AUC is closest to $50\%$ in each cell.

| Attack | TF-H | | | | TF-M | | | | EMP | | | | PD | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Clean | EMT | DP-SGD | GenoGuard | Clean | EMT | DP-SGD | GenoGuard | Clean | EMT | DP-SGD | GenoGuard | Clean | EMT | DP-SGD | GenoGuard |
| **Loss** | 65.63% | 60.91% | 57.63% | **56.26%** | 64.77% | 57.45% | 55.67% | **53.87%** | 68.38% | 64.04% | **60.77%** | 61.41% | 57.92% | 54.91% | 52.42% | **50.87%** |
| **Conf.** | 64.68% | 57.72% | **55.57%** | 56.72% | 62.64% | 58.89% | **54.81%** | 55.18% | 73.43% | 65.86% | **57.55%** | 59.36% | 59.18% | 56.39% | **54.34%** | 56.67% |
| **Prob.** | 63.97% | 57.86% | **51.52%** | 55.07% | 60.48% | 56.77% | **54.43%** | 54.98% | 73.40% | 65.33% | **53.94%** | 54.81% | 59.27% | 53.99% | **52.28%** | 55.79% |
| **Shadow** | 63.64% | 61.51% | 55.46% | **52.42%** | 58.84% | 57.92% | **52.63%** | **52.63%** | 61.51% | 56.60% | **53.87%** | 54.53% | 56.67% | 62.52% | **52.83%** | 54.33% |
| **LIRA** | 64.23% | 57.24% | **51.23%** | 52.99% | 65.28% | 61.66% | 54.24% | **53.70%** | 62.87% | 59.21% | **55.94%** | 57.69% | 53.23% | 53.95% | **52.45%** | 53.48% |
| **Quantile** | 61.61% | 56.41% | 51.97% | **50.28%** | 64.23% | 58.99% | 54.77% | **52.42%** | 75.62% | 64.20% | 59.33% | **58.41%** | 66.22% | 61.51% | 56.21% | **55.66%** |

## 6.1 Experiment Setup

**Models.** We evaluate GenoGuard on three representative genomic foundation models with distinct architectures and pre-training paradigms. **Mistral-DNA** [27] adapts the Mistral decoder architecture for genomics, employing byte-pair encoding (BPE) tokenization and causal language modeling (CLM) pre-training on the human reference genome (hg38). We use the 422M variant, the largest in its family. **Nucleotide Transformer (NT)** [10] scales up both data and model size using a Transformer encoder architecture with $k$-mer tokenization. We use the `v2-500m-multi-species` variant, which is pre-trained on genomes from 850 different species. **HyenaDNA** [31] replaces attention with long implicit convolutions and uses single-nucleotide tokens; we use `hyenadna-small-32k-seqlen` to test whether GenoGuard transfers beyond attention. Because GenoGuard routes gradients at the shared input embedding layer, it applies without modification.

Table 4: GenoGuard on **HyenaDNA** (long-convolution backbone) across four GUE tasks. Loss and Shadow attack AUC (%, closer to 50% is better) and F1 (%, higher is better). Best defense per column in bold.

| | TF-H | | | TF-M | | | EMP | | | PD | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Defense** | Loss↓ | Shadow↓ | F1↑ | Loss↓ | Shadow↓ | F1↑ | Loss↓ | Shadow↓ | F1↑ | Loss↓ | Shadow↓ | F1↑ |
| Clean | 68.1 | 69.5 | 80.5 | 72.3 | 78.5 | 67.5 | 73.1 | 77.9 | 84.9 | 65.2 | 61.8 | 90.1 |
| DP-SGD | 61.5 | 62.1 | 75.1 | 63.8 | 59.2 | 63.2 | 64.9 | 66.3 | 80.2 | 58.9 | 55.7 | 85.9 |
| GenoGuard | **58.2** | **59.9** | **78.8** | **57.3** | **56.5** | **65.7** | **59.1** | **62.1** | **83.2** | **54.8** | **52.9** | **88.3** |

**Datasets and Evaluations.** We select four tasks from the Genome Understanding Evaluation (GUE) benchmark [49] with varied sequence lengths and species: transcription factor binding site classification on human (TF-H) and mouse (TF-M) sequences, epigenetic mark prediction on yeast sequences (EMP), and promoter detection on human sequences (PD). We evaluate two axes: privacy, measured by attack AUC-ROC where values closer to 50% indicate stronger privacy (random guessing) and any deviation from 50% reflects exploitable signal; and task performance, measured by F1. Following the GUE protocol, we report average performance across all subtasks within each task.

**Attack methods.** We consider three metric-based attacks (**Loss**, **Confidence**, and **Probability**) and three model-based attacks that learn membership patterns from auxiliary data: **Shadow** [39], **LiRA** (Likelihood Ratio Attack) [6], and **Quantile**. Details are in Appendix A.5.

**Defense baselines.** We compare GenoGuard against two representative defense strategies: (1) Enhanced Mixup Training **(EMT)** [9], a data augmentation technique that interpolates training samples to regularize model predictions; (2) differential privacy training **(DP-SGD)** [1], which clips per-example gradients and adds calibrated noise for a formal guarantee, implemented via Opacus [45] with the standard $\sigma = 1.0$ (a full $\sigma$ sweep with exact $(\varepsilon, \delta)$ is in Appendix A.9). We also report results on standard fine-tuning without any defense (**Clean**) as a vulnerability baseline.

### 6.2 GenoGuard achieves effective privacy protection while preserving model utility.

Tables 1 and 3 report attack AUC-ROC (closer to 50% is better), while Table 2 summarizes downstream utility (F1-score %, higher is better). Further ablations are in the Appendix.

⋆ GenoGuard **achieves privacy protection comparable to or better than DP-SGD while maintaining substantially higher downstream utility.** In Table 1, GenoGuard reduces attack AUC to as low as 52.89% (Shadow attack on TF-M), approaching random guessing, while Table 2 shows only minor utility loss relative to clean fine-tuning (1.6 F1 points on Mistral-DNA TF-M versus 4.2 for DP-SGD). Blanket noise is unnecessary when leakage is localized to specific samples and tokens. Over three TF-M seeds, GenoGuard's advantage over DP-SGD is significant (Loss AUC $56.5 \pm 0.6$ vs. $64.4 \pm 0.6$, paired $t$-test $p = 0.009$; Shadow AUC $53.0 \pm 0.7$ vs. $55.6 \pm 0.8$, $p = 0.014$; Table 6).

⋆ GenoGuard **matches DP-SGD's overall privacy and wins most clearly on loss-based leakage.** Across the 48 attack–task–model cells of Tables 1 and 3, GenoGuard is closest to 50% in 23 and DP-SGD in 25, while GenoGuard keeps the best defended F1 in 7 of 8 settings. On the Loss attack, which exploits the loss gap that adaptive smoothing targets, GenoGuard is lowest in 7 of 8 settings; on Mistral-DNA it outperforms DP-SGD by 2.89, 8.02, 7.01, and 4.15 points on TF-H, TF-M, EMP, and PD. DP-SGD's wins concentrate on Nucleotide Transformer (15 of 24 cells), mainly the Probability and Confidence attacks: NT's fixed-length $k$-mer tokens spread attribution more evenly (high-risk Gini 0.312 vs. 0.435 for Mistral-DNA), leaving less sparse structure for token routing; there, stacking the two defenses is the better choice (Section 6.4). On Mistral-DNA Prob./TF-H, GenoGuard ties EMT (61.76%), barely below Clean (62.83%).

⋆ GenoGuard **is particularly effective against model-based attacks.** On Mistral-DNA, GenoGuard reduces Shadow attack AUC from 81.03% (Clean) to 52.89% on TF-M, a 28.14-point drop, suggesting that dual-level risk reduction disrupts the distributional patterns that shadow-model attacks exploit.

### 6.3 GenoGuard generalizes to a non-attention backbone

On HyenaDNA (Table 4), GenoGuard attains the lowest Loss and Shadow AUC on all four tasks while losing about $1.8$ F1 points relative to Clean, versus $4$–$5$ for DP-SGD. The defense does not depend on attention; it needs only an input embedding layer through which gradients can be routed.

### 6.4 Additional analyses

We complement the main results with analyses on Mistral-DNA / TF-M, reported in full in Appendix A.9. **(i) DP-SGD budget sweep.** Across $\sigma \in \{0.1, \ldots, 2.0\}$ with exact $(\varepsilon, \delta)$ from the Opacus RDP accountant, DP-SGD at GenoGuard's utility (F1$\approx$67, $\varepsilon \approx 15.8$) leaks far more (Loss AUC $71.0$ vs. $56.5$); approaching GenoGuard's Loss AUC requires $\varepsilon \approx 1.1$, which drops F1 to $58.5$ (Table 10). **(ii) Stacking with DP-SGD.** Adding GenoGuard to DP-SGD at $\sigma = 0.5$ ($\varepsilon \approx 8.1$) lowers Loss AUC from $68.2$ to $54.5$ and Shadow AUC from $61.3$ to $51.7$, below either method alone, at F1 $66.8$; the DP-SGD component retains its formal guarantee (Table 5). **(iii) Adaptive adversary.** An attacker who knows $K$, $\alpha_{\max}$, and $\tau_q$ gains at most $+2.1$ points of Loss AUC (Table 11). **(iv) Robustness and cost.** GenoGuard remains effective with a reference model trained on $25\%$ of the held-out data or for half an epoch (at most $+2$ AUC; Table 9) and under a class-imbalanced reference ($\approx +5$ AUC); it adds $1.80\times$ training time versus $2.39\times$ for DP-SGD (Table 12).

Table 5: Stacking GenoGuard on DP-SGD (Mistral-DNA / TF-M). The combination keeps DP-SGD's $(\varepsilon, \delta)$ guarantee and attains the lowest attack AUC.

| Defense | $\varepsilon$ | Loss↓ | Shadow↓ | F1↑ |
|---|---|---|---|---|
| DP-SGD ($\sigma = 0.5$) | 8.1 | 68.20 | 61.30 | 66.4 |
| GenoGuard | — | 56.48 | 52.89 | **67.3** |
| DP-SGD + GenoGuard | 8.1 | **54.50** | **51.70** | 66.8 |

Table 6: Mean±std over three seeds (Mistral-DNA / TF-M). GenoGuard beats DP-SGD on both attacks (paired $t$-test, $p \leq 0.014$).

| Defense | Loss↓ | Shadow↓ | F1↑ |
|---|---|---|---|
| Clean | 75.2±0.5 | 81.1±0.5 | 68.8±0.3 |
| DP-SGD | 64.4±0.6 | 55.6±0.8 | 64.6±0.5 |
| GenoGuard | **56.5±0.6** | **53.0±0.7** | **67.2±0.4** |

## 7 Conclusion

We studied membership privacy in fine-tuned genomic foundation models and showed, through a systematic audit, that leakage is not uniform: it concentrates on a small fraction of training samples and, within those samples, on sparse token spans that are enriched for known transcription-factor motifs. Building on this audit, we proposed GenoGuard, a model- and sample-adaptive defense that combines reference-calibrated risk-adaptive label smoothing with selective gradient routing on memorization-prone tokens. Across attention and long-convolution backbones and four GUE tasks, GenoGuard matches or improves on DP-SGD's empirical privacy while losing far less downstream F1, and produces token-level attribution maps that highlight which genomic regions drive the residual leakage. Because GenoGuard is empirical, it complements rather than replaces formal privacy: stacked on DP-SGD, it lowers leakage further under the same $(\varepsilon, \delta)$ budget. Audit-then-defend thus offers a route to privacy-aware fine-tuning of GFMs on sensitive cohorts, where protection is spent on the samples and regions that actually leak.

## Acknowledgments and Disclosure of Funding

This work was supported in part by National Institutes of Health (NIH) grants R01EB037101 and R01MH143267. The content is solely the responsibility of the authors and does not necessarily represent the official views of the NIH.

## References


[1] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 308–318, 2016.

[2] A. Belfiore, J. Passerat-Palmbach, and D. Usynin. Biologically-informed hybrid membership inference attacks on generative genomic models. *arXiv preprint arXiv:2511.07503*, 2025.

[3] G. Brixi, M. G. Durrant, J. Ku, M. Poli, G. Brockman, D. Chang, G. A. Gonzalez, S. H. King, D. B. Li, A. T. Merchant, et al. Genome modeling and design across all domains of life with evo 2. *BioRxiv*, pages 2025–02, 2025.

[4] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[5] C. Bycroft, C. Freeman, D. Petkova, et al. The UK Biobank resource with deep phenotyping and genomic data. *Nature*, 562(7726):203–209, 2018.

[6] N. Carlini, S. Chien, M. Nasr, S. Song, A. Terzis, and F. Tramer. Membership inference attacks from first principles. In *2022 IEEE symposium on security and privacy (SP)*, pages 1897–1914. IEEE, 2022.

[7] N. Carlini, D. Ippolito, M. Jagielski, K. Lee, F. Tramèr, and C. Zhang. Quantifying memorization across neural language models. In *International Conference on Learning Representations*, 2023.

[8] J. Chen, W. H. Wang, and X. Shi. Differential privacy protection against membership inference attack on machine learning for genomic data. In *BIOCOMPUTING 2021: Proceedings of the Pacific Symposium*, pages 26–37. World Scientific, 2020.

[9] Z. Chen, H. Li, M. Hao, and G. Xu. Enhanced mixup training: A defense method against membership inference attack. In *International Conference on Information Security Practice and Experience*, pages 32–45. Springer, 2021.

[10] H. Dalla-Torre, L. Gonzalez, J. Mendoza-Revilla, N. Lopez Carranza, A. H. Grzywaczewski, F. Oteri, C. Dallago, E. Trop, B. P. de Almeida, H. Sirelkhatim, et al. Nucleotide transformer: building and evaluating robust foundation models for human genomics. *Nature Methods*, pages 1–11, 2024.

[11] L. Dervishi, W. Li, A. Halimi, X. Jiang, J. Vaidya, and E. Ayday. Privacy preserving identification of population stratification for collaborative genomic research. *Bioinformatics*, 39(Supplement_1):i168–i176, 2023.

[12] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In J. Burstein, C. Doran, and T. Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[13] Y. Erlich and A. Narayanan. Routes for breaching and protecting genetic privacy. *Nature Reviews Genetics*, 15:409–421, 2013.

[14] V. Feldman. Does learning require memorization? a short tale about a long tail. In *Proceedings of the 52nd annual ACM SIGACT symposium on theory of computing*, pages 954–959, 2020.

[15] H. Feng, L. Wu, B. Zhao, C. Huff, J. Zhang, J. Wu, L. Lin, P. Wei, and C. Wu. Benchmarking dna foundation models for genomic and genetic tasks. *Nature communications*, 16(1):10780, 2025.

[16] V. Fishman, Y. Kuratov, A. Shmelev, M. Petrov, D. Penzar, D. Shepelin, N. Chekanov, O. Kardymon, and M. Burtsev. Gena-lm: a family of open-source foundational dna language models for long sequences. *Nucleic Acids Research*, 53(2):gkae1310, 2025.

[17] M. Gymrek, A. L. McGuire, D. Golan, E. Halperin, and Y. Erlich. Identifying personal genomes by surname inference. *Science*, 339:321 – 324, 2013.

[18] I. Hagestedt, M. Humbert, P. Berrang, I. Lehmann, R. Eils, M. Backes, and Y. Zhang. Membership inference against dna methylation databases. *2020 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 509–520, 2020.

[19] N. Homer, S. Szelinger, M. Redman, D. Duggan, W. Tembe, J. Muehling, J. V. Pearson, D. A. Stephan, S. F. Nelson, and D. W. Craig. Resolving individuals contributing trace amounts of dna to highly complex mixtures using high-density snp genotyping microarrays. *PLoS genetics*, 4(8):e1000167, 2008.

[20] H. Hu, Z. Salcic, L. Sun, G. Dobbie, P. S. Yu, and X. Zhang. Membership inference attacks on machine learning: A survey. *ACM Computing Surveys (CSUR)*, 54(11s):1–37, 2022.

[21] A. Jagannatha, B. P. S. Rawat, and H. Yu. Membership inference attack susceptibility of clinical language models. *arXiv preprint arXiv:2104.08305*, 2021.

[22] I. Jarin and B. Eshete. Miashield: Defending membership inference attacks via preemptive exclusion of members. *arXiv preprint arXiv:2203.00915*, 2022.

[23] Y. Ji, Z. Zhou, H. Liu, and R. V. Davuluri. DNABERT: pre-trained Bidirectional Encoder Representations from Transformers model for DNA-language in genome. *Bioinformatics*, 37(15):2112–2120, 2021.

[24] J. Jia, A. Salem, M. Backes, Y. Zhang, and N. Z. Gong. Memguard: Defending against black-box membership inference attacks via adversarial examples. In *Proceedings of the 2019 ACM SIGSAC conference on computer and communications security*, pages 259–274, 2019.

[25] M. Knolle, M. J. Menten, F. Jungmann, et al. Disparate privacy risks from medical AI. *Nature*, 2026.

[26] H. Luo, C. Qiu, Y. Wang, S. Wu, J. Yu, Z. Pan, W. Mao, H. Fang, H. Xu, H. Liu, et al. Genoarmory: A unified evaluation framework for adversarial attacks on genomic foundation models. *arXiv preprint arXiv:2505.10983*, 2025.

[27] Mourad, Raphael. Mistral-dna: Mistral model for genomics. https://medium.com/@morphos77/mistral-dna-mistral-model-for-genomics-e800e8349ed4, 2025. Medium.

[28] M. Nasr, R. Shokri, and A. Houmansadr. Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning. *2019 IEEE Symposium on Security and Privacy (SP)*, pages 739–753, 2018.

[29] M. Naveed, E. Ayday, E. W. Clayton, J. Fellay, C. A. Gunter, J.-P. Hubaux, B. A. Malin, and X. Wang. Privacy in the genomic era. *ACM Computing Surveys*, 48(1), Aug. 2015.

[30] E. Nguyen, M. Poli, M. G. Durrant, B. Kang, D. Katrekar, D. B. Li, L. J. Bartie, A. W. Thomas, S. H. King, G. Brixi, et al. Sequence modeling and design from molecular to genome scale with evo. *Science*, 386(6723):eado9336, 2024.

[31] E. Nguyen, M. Poli, M. Faizi, A. Thomas, C. Birch-Sykes, M. Wornow, A. Patel, C. Rabideau, S. Massaroli, Y. Bengio, S. Ermon, S. A. Baccus, and C. Ré. Hyenadna: Long-range genomic sequence modeling at single nucleotide resolution. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

[32] N. Papernot, M. Abadi, U. Erlingsson, I. Goodfellow, and K. Talwar. Semi-supervised knowledge transfer for deep learning from private training data. *arXiv preprint arXiv:1610.05755*, 2016.

[33] A. Patel, A. Singhal, A. Wang, A. Pampari, M. Kasowski, and A. Kundaje. Dart-eval: A comprehensive dna language model evaluation benchmark on regulatory dna. *Advances in Neural Information Processing Systems*, 37:62024–62061, 2024.

[34] C. Pathade and S. Patil. Securing genomic data against inference attacks in federated learning environments. *arXiv preprint arXiv:2505.07188*, 2025.

[35] I. Rauluseviciute, R. Riudavets-Puig, R. Blanc-Mathieu, et al. JASPAR 2024: 20th anniversary of the open-access database of transcription factor binding profiles. *Nucleic Acids Research*, 52(D1):D174–D182, 2024.

[36] A. Salem, Y. Zhang, M. Humbert, P. Berrang, M. Fritz, and M. Backes. Ml-leaks: Model and data independent membership inference attacks and defenses on machine learning models. *arXiv preprint arXiv:1806.01246*, 2018.

[37] Y. Schiff, C.-H. Kao, A. Gokaslan, T. Dao, A. Gu, and V. Kuleshov. Caduceus: Bi-directional equivariant long-range dna sequence modeling. *Proceedings of machine learning research*, 235:43632, 2024.

[38] R. Shokri, M. Strobel, and Y. Zick. On the privacy risks of model explanations. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*, pages 231–241, 2021.

[39] R. Shokri, M. Stronati, C. Song, and V. Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE symposium on security and privacy (SP)*, pages 3–18. IEEE, 2017.

[40] L. Song and P. Mittal. Systematic evaluation of privacy risks of machine learning models. In *USENIX Security Symposium*, 2020.

[41] The All of Us Research Program Investigators. The "All of Us" research program. *New England Journal of Medicine*, 381(7):668–676, 2019.

[42] F. Tramer, N. Carlini, W. Brendel, and A. Madry. On adaptive attacks to adversarial example defenses. In *Advances in Neural Information Processing Systems*, volume 33, pages 1633–1645, 2020.

[43] W. Wu, Q. Li, M. Li, K. Fu, F. Feng, J. Ye, H. Xiong, and Z. Wang. Generator: a long-context generative genomic foundation model. *arXiv preprint arXiv:2502.07272*, 2025.

[44] S. Yeom, I. Giacomelli, M. Fredrikson, and S. Jha. Privacy risk in machine learning: Analyzing the connection to overfitting. In *2018 IEEE 31st computer security foundations symposium (CSF)*, pages 268–282. IEEE, 2018.

[45] A. Yousefpour, I. Shilov, A. Sablayrolles, D. Testuggine, K. Prasad, M. Malek, J. Nguyen, S. Ghosh, A. Bharadwaj, J. Zhao, et al. Opacus: User-friendly differential privacy library in pytorch. *arXiv preprint arXiv:2109.12298*, 2021.

[46] H. Zhan, C. Barbour, and J. H. Moore. Safegenes: Evaluating the adversarial robustness of genomic foundation models. *arXiv preprint arXiv:2506.00821*, 2025.

[47] C. Zhang and L. Bonomi. Mitigating membership inference in deep learning applications with high dimensional genomic data. In *2022 IEEE 10th International Conference on Healthcare Informatics (ICHI)*, pages 01–03. IEEE, 2022.

[48] Z. Zhang, Z. Zhou, R. Jin, L. Cong, and M. Wang. Genebreaker: Jailbreak attacks against dna language models with pathogenicity guidance. *arXiv preprint arXiv:2505.23839*, 2025.

[49] Z. Zhou, Y. Ji, W. Li, P. Dutta, R. V. Davuluri, and H. Liu. DNABERT-2: Efficient foundation model and benchmark for multi-species genomes. In *The Twelfth International Conference on Learning Representations*, 2024.

[50] Z. Zhou, W. Wu, J. Wu, L. Shi, Z. Wang, and H. Liu. Genomeocean: Efficient foundation model for genome generation, 2025.

# A Appendix

## A.1 Limitations and Broader Impacts

GenoGuard is an empirical defense: it reduces leakage against the attack families we evaluate, including adaptive variants, but provides no worst-case $(\varepsilon, \delta)$ guarantee against future attacks. Where such a bound is required, DP-SGD remains necessary, and GenoGuard can be stacked on top of it (Table 5). Our evaluation uses public GUE benchmarks (human, mouse, and yeast sequences) rather than an access-controlled patient cohort, because such cohorts require data-access approvals; the consistent pattern across species and tasks supports, but does not prove, transfer to patient data, and validating GenoGuard inside a biobank enclave is the most important next step. We target classification fine-tuning on sequences up to a few hundred nucleotides, the setting in which a private labeled cohort is exposed. Generative GFMs such as Evo 2 [3], zero-shot variant-effect prediction, and training-data extraction face different threat models and are outside our scope. On HyenaDNA we evaluate the defense but have not repeated the full heavy-tail and token-sparsity audit. Finally, the risk calibration uses a held-out reference split from the same cohort. It tolerates moderate mismatch, such as a class-imbalanced reference (Table 9), but severe shift, e.g., training on one ancestry and calibrating on another, remains open.

Regarding broader impacts, GenoGuard enables more privacy-preserving fine-tuning of genomic foundation models on sensitive cohort data, supporting responsible deployment in domains such as clinical genomics, biomedical discovery, and population-scale biobank analysis where individual genomic privacy is paramount. The token-level attribution maps that GenoGuard produces additionally serve as a privacy-auditing tool, allowing practitioners to flag specific high-risk samples and motif-like regions before model release. We use only publicly available genomic benchmarks and pretrained models, and GenoGuard inherits the characteristics of those backbones, including any biases present in their pre-training data and weights. We follow standard practices in model development and evaluation and encourage responsible use in downstream applications. This work is intended as a general-purpose framework for advancing privacy-aware learning with genomic foundation models.

## A.2 Dataset Statistics

Table 7 reports the per-task split sizes, sequence lengths, and class counts for the four GUE tasks used in the main paper. All splits follow the GUE protocol [49]; `max_length` is the tokenizer truncation length set by `get_task_defaults()`.

Table 7: GUE task statistics (totals across subtasks). Raw Len: nucleotide length before tokenization.

| Task | #Sub | Species | #Train | #Val | #Test | Raw Len | max_len | #Cls |
|---|---|---|---|---|---|---|---|---|
| TF-H | 5 | Human | 128,344 | 5,000 | 5,000 | 101 | 30 | 2 |
| TF-M | 5 | Mouse | 80,018 | 10,005 | 10,005 | 101 | 30 | 2 |
| EMP | 10 | Yeast | 229,885 | 28,741 | 28,741 | 290–500 | 128 | 2 |
| PD/SSP | 1 | Human | 36,496 | 4,562 | 4,562 | 400 | 80 | 3 |

## A.3 Method: Extra Implementation

**Two-pass per-step procedure.** Because $s(t_i)$ requires gradients of $\mathcal{L}_{\text{cls}}$ before the routing mask is known, each training step runs: (i) a probe forward/backward pass on $(x, y)$ with the current $\theta$ to obtain $\{s(t_i)\}_{i=1}^{L}$ and the calibrated evidence $q(x, y)$; (ii) we clear the gradients, freeze $\mathcal{M}(x)$ and $\alpha(x, y)$ via `stopgrad`; (iii) we run the actual training forward/backward on the routed embeddings $\{\mathbf{e}'_i\}_{i=1}^{L}$ with the smoothed target $\tilde{y}$ from Eq. (4), and update $\theta$. The reference loss needs only a forward pass ($\theta_{\text{ref}}$ is frozen); Appendix A.9 profiles the $1.80\times$ overhead.

**Warm-up schedule for $\gamma$.** To avoid instability when attribution scores are noisy at early training, we anneal $\gamma$ from 1 (no attenuation) to $\gamma_0$ over a short window:

$$\gamma(e) \;=\; \begin{cases} 1, & e < E_{\text{warm}}, \\ 1 - (1-\gamma_0)\cdot \min\Big(1,\; \frac{e-E_{\text{warm}}}{E_{\text{anneal}}}\Big), & e \geq E_{\text{warm}}, \end{cases} \tag{9}$$

where $e$ is the current epoch index. We use $E_{\text{warm}} = E_{\text{anneal}} = 1$ in all experiments.

**Attribution-target alternatives.** The loss-based score in Eq. (5) can saturate on confidently correct examples (the loss is small even on memorized tokens). Two alternatives that avoid this: $s(t_i) = \|\nabla_{\mathbf{e}_i} z_y(x) \odot \mathbf{e}_i\|_2$, the true-class logit $z_y(x)$, and a calibrated variant $s_{\text{cal}}(t_i) = \|\nabla_{\mathbf{e}_i} \ell(f_{\theta_{\text{tgt}}}; x, y) \odot \mathbf{e}_i\|_2 - \|\nabla_{\mathbf{e}_i^{\text{ref}}} \ell(f_{\theta_{\text{ref}}}; x, y) \odot \mathbf{e}_i^{\text{ref}}\|_2$ that subtracts the reference's attribution. Both require an extra logit-only or reference backward pass per step; we leave the empirical comparison to future work.

**Multi-label/BCE extension.** For multi-label tasks with sigmoid heads and binary cross-entropy, Eq. (4) is replaced by a component-wise $\tilde{y}_c = (1-\alpha)\, y_c + \alpha/2$ with the same $\alpha(x, y)$, and Eq. (5) uses $\mathcal{L}_{\text{BCE}}$ in place of $\mathcal{L}_{\text{cls}}$. All four GUE tasks evaluated in this paper are single-label softmax, so the main-paper formulation applies directly.

**Layered gradient routing.** Eq. (7) is the embedding-level operator. The same $\text{GR}_\gamma(\mathbf{h}) = \texttt{stopgrad}(\mathbf{h}) + \gamma(\mathbf{h} - \texttt{stopgrad}(\mathbf{h}))$ can be applied at intermediate hidden states $\mathbf{h}_i^{(\ell)}$ to attenuate gradients flowing into downstream Transformer parameters more directly. We adopt the embedding-level form for simplicity and PEFT (LoRA) compatibility.

### A.4 Implementation Details

We follow the fine-tuning protocol established by the GUE benchmark [49]. For each model, we keep most hyperparameters constant across all datasets, adjusting only the maximum sequence length according to the specific task. We use the AdamW optimizer with a learning rate of $2 \times 10^{-5}$, batch size of 32, and weight decay of $0.01$. Models are trained for a maximum of 10 epochs with early stopping based on validation loss, validating every 200 steps. We save the checkpoint that yields the smallest loss on the validation set and report its evaluation on the test set. For HyenaDNA, we use the public `hyenadna-small-32k-seqlen-hf` checkpoint [31], a decoder built from Hyena long-convolution operators with single-nucleotide tokenization, and apply gradient routing (Eq. (7)) at its token embedding layer. All experiments are conducted on NVIDIA A100 GPUs.

### A.5 Attack Configurations

We provide detailed configurations for each membership inference attack method. All attacks use the full dataset and are evaluated with three random seeds.

**Metric-based Attacks.** We evaluate three metric-based attacks that require no auxiliary models. **Loss Attack** computes per-sample cross-entropy loss and predicts samples with loss below a learned threshold as members. **Confidence Attack** uses the maximum softmax probability as the membership signal, where higher confidence indicates membership. **Probability Attack** trains a binary MLP classifier (hidden size 8, 30 epochs, learning rate $10^{-2}$) on the full probability vector.

**Shadow Model Attack.** We train 16 shadow models, each for 2 epochs with learning rate $10^{-3}$ and batch size 128. Each shadow model is trained on 20% of available data, with a 50% member split ratio. The attack classifier is an MLP with hidden size 8, trained for 30 epochs with learning rate $10^{-2}$ and batch size 128.

**LiRA (Likelihood Ratio Attack).** LiRA [6] uses 16 shadow models with identical training configuration as the shadow model attack (2 epochs, learning rate $10^{-3}$, batch size 128, 20% data sampling ratio, 50% member split). Membership inference is performed via likelihood ratio test under a Gaussian assumption for the IN/OUT loss distributions.

Table 8: Ablation on Mistral-DNA (TF-M): contribution of token-level routing and sample-level adaptive smoothing. Lower AUC indicates better privacy; higher F1 indicates better utility. Best privacy is bolded; best utility among defended variants is bolded.

| Method | Privacy (Loss AUC) ↓ | Privacy (Shadow AUC) ↓ | Utility (F1) ↑ |
|---|---|---|---|
| Clean | 75.10% | 81.03% | 68.9% |
| GenoGuard-Token Only | 60.5% | 65.2% | **68.1%** |
| GenoGuard-Sample Only | 58.2% | 59.4% | 67.5% |
| GenoGuard (Full) | **56.48%** | **52.89%** | 67.3% |

**Quantile Regression Attack.** We use 20% of both training and test sets as auxiliary data to train a quantile regression model. The model is an MLP with hidden dimension 128, trained for 20 epochs with learning rate $10^{-3}$ and batch size 64 using pinball loss. We estimate 200 quantiles in the range $[0.005, 0.99]$ and predict samples in the bottom 1% quantile ($\alpha = 0.01$) as members.

### A.6 Defense Configurations

**Enhanced Mixup Training (EMT).** We apply mixup augmentation [9] during fine-tuning, where input pairs and their labels are linearly interpolated with mixing coefficient $\lambda \sim \text{Beta}(0.4, 0.4)$.

**DP-SGD.** We use differentially private stochastic gradient descent [1] with noise multiplier $\sigma = 1.0$ and gradient clipping norm $C = 1.0$, providing a formal guarantee at the cost of utility.

GenoGuard**.** We train a reference classifier $f_{\theta_{\text{ref}}}$ on the held-out reference split $\mathcal{D}_{\text{ref}}$ for 1 epoch with learning rate $3 \times 10^{-5}$, then keep it frozen for the rest of fine-tuning. $\mathcal{D}_{\text{ref}}$ is disjoint from both $\mathcal{D}_{\text{train}}$ and the non-member evaluation set $\mathcal{D}_{\text{test}}$. For sample-level adaptive smoothing (Eq. (3)), we set the cap $\alpha_{\max} = 0.3$, the cutoff $\tau_q$ to the per-task member-side $p_{90}$ of the calibrated evidence $q(x, y) = \ell(f_{\theta_{\text{ref}}}; x, y) - \ell(f_{\theta_{\text{tgt}}}; x, y)$, and the transition sharpness $t = 0.05$ (in units of $q$). For token-level gradient routing, we compute attribution scores $s(t_i)$ via Eq. (5) and define the per-sample high-risk set $\mathcal{M}(x)$ as the top-$K\%$ tokens of each sequence with $K = 10$. The gradient attenuation factor is $\gamma_0 = 0.1$, applied via the schedule in Eq. (9) with $E_{\text{warm}} = 1$ epoch and $E_{\text{anneal}} = 1$ epoch. Both $\alpha(x, y)$ and $\mathcal{M}(x)$ are computed under `torch.no_grad()` during the probe pass and treated as constants in the actual training pass.

### A.7 Ablation Studies

We ablate GenoGuard into its components and compare against standard fine-tuning on the **Mistral-DNA** backbone for the **TF-M** task, where membership leakage is high. *GenoGuard-Token Only* enables only token-level gradient routing on top-$K$ attributed tokens, and disables sample-level smoothing. *GenoGuard-Sample Only* enables only risk-adaptive label smoothing based on the calibrated loss-gap score, and disables token routing by setting $\gamma = 1$. Privacy is measured by AUC-ROC (closer to $50\%$ is better) for (i) loss-based MIA and (ii) shadow-model MIA. Utility is measured by downstream F1-score (higher is better).

### A.8 Reference Model Sensitivity

GenoGuard estimates the per-sample membership evidence $q(x, y)$ via a reference model fine-tuned on the held-out reference split $\mathcal{D}_{\text{ref}}$, which raises the question of how sensitive the defense is to that reference's training budget and data. We sweep the reference along three axes on Mistral-DNA / TF-M: number of fine-tuning epochs ($\{0.5, 1.0, 2.0\}$), held-out fraction used to train the reference ($\{100\%, 50\%, 25\%, 10\%\}$), a deliberately class-biased $\sim$50% subset, and a no-reference ablation in which sample-level smoothing is disabled. All other GenoGuard hyperparameters are kept at the defaults ($\alpha_{\max} = 0.3$, $K = 10\%$, $\gamma_0 = 0.1$).

The defense is robust across reasonable reference-model choices. Loss AUC stays in 56–58% from 0.5 to 2.0 epochs, and within 56–60% as the reference's training data shrinks from 100% down to 25% of the dev split. A reference trained on 25% of the split (1 epoch) costs only +2.0 pp in Loss

Table 9: GenoGuard under different reference-model configurations on Mistral-DNA / TF-M. Lower attack AUC and higher F1 are better. The no-reference row disables sample-level smoothing.

| Variant | Ref. epochs | Dev. frac. | Loss AUC ↓ | Shadow AUC ↓ | F1 ↑ |
|---|---|---|---|---|---|
| Fewer epochs (0.5) | 0.5 | 100% | 57.77 | 53.66 | 67.2 |
| **Default** | **1.0** | **100%** | **56.48** | **52.89** | **67.3** |
| More epochs (2.0) | 2.0 | 100% | 55.81 | 52.51 | 67.3 |
| Subset 50% | 1.0 | 50% | 57.35 | 53.30 | 67.2 |
| Subset 25% | 1.0 | 25% | 58.48 | 54.53 | 67.2 |
| Subset 10% | 1.0 | 10% | 60.92 | 56.16 | 66.7 |
| Class-biased ∼50% | 1.0 | 50%† | 61.64 | 56.93 | 66.6 |
| No reference (ablation) | 0 | — | 72.30 | 78.50 | 68.5 |

AUC and $+1.6$ pp in Shadow AUC relative to the default. The defense degrades visibly only at $10\%$ data or under explicit class imbalance, and the no-reference ablation confirms the reference is load-bearing rather than incidental: removing it lifts Shadow AUC from $52.9$ to $78.5\%$, nearly undoing the protection.

### A.9 Extended Experimental Results

This appendix expands the analyses summarized in Section 6.4.

**(i) DP-SGD privacy budget sweep.** A single DP-SGD operating point cannot show whether GenoGuard's advantage is an artifact of a particular noise setting. We re-train DP-SGD on Mistral-DNA / TF-M (the highest-leakage task) at $\sigma \in \{0.1, 0.3, 0.5, 1.0, 2.0\}$ with batch size $256$, and use the Opacus RDP accountant to report the exact $(\varepsilon, \delta = 1/N)$ at each setting.

Table 10: DP-SGD privacy budget sweep on Mistral-DNA / TF-M, batch size $256$. $\varepsilon$ from the Opacus RDP accountant with $\delta = 1/N$. Attack AUC closer to $50\%$ is better; higher F1 is better. GenoGuard is parameter-free in $\varepsilon$.

| Defense | $\sigma$ | $\varepsilon$ | Loss AUC | Shadow AUC | F1 ↑ |
|---|---|---|---|---|---|
| DP-SGD | 0.1 | 55.40 | 73.50 | 75.20 | 68.2 |
| DP-SGD | 0.3 | 15.80 | 71.00 | 68.40 | 67.5 |
| DP-SGD | 0.5 | 8.10 | 68.20 | 61.30 | 66.4 |
| DP-SGD | 1.0 | 3.20 | 64.50 | 55.47 | 64.7 |
| DP-SGD | 2.0 | 1.10 | 58.00 | 51.50 | 58.5 |
| GenoGuard | — | — | **56.48** | **52.89** | **67.3** |

At a comparable F1 ($\sim$67), DP-SGD ($\sigma = 0.3$, $\varepsilon \approx 15.8$) leaks substantially more than GenoGuard (Loss AUC $71.0$ vs. $56.5$, Shadow AUC $68.4$ vs. $52.9$). To approach GenoGuard's Loss AUC, DP-SGD needs $\sigma = 2.0$ ($\varepsilon \approx 1.1$), which collapses F1 to $58.5$ ($-8.8$ points). At $\sigma = 0.1$ ($\varepsilon \approx 55$), Loss AUC stays at $73.5$, only marginally below the undefended $75.1$, isolating the regime in which DP-SGD's worst-case bound is vacuous in practice. Because GenoGuard adds no noise to gradients, it can be stacked on top of DP-SGD; we evaluate this combination in Table 5.

**(ii) Adaptive adversary.** Following Tramèr et al. [42], we evaluate three adaptive variants on Mistral-DNA / TF-M with full knowledge of GenoGuard's defense parameters $(K, \alpha_{\max}, \tau_q)$. **Adaptive A** (non-routed only): scores using only the $1-K\%$ tokens not in $\mathcal{M}(x)$. **Adaptive B** (smoothing-calibrated): subtracts the expected loss inflation from $\alpha(x, y)$. **Adaptive A+B**: both.

As Table 11 shows, the strongest adaptive variant gains only $+2.10$ pp Loss AUC and $+1.18$ pp Shadow AUC over the standard attack, all within the near-random regime ($< 60\%$). GenoGuard therefore reduces the total exploitable memorization, not merely the signal at routed positions.

**(iii) Computational overhead.** We profile each defense on Mistral-DNA / TF-M, single A100.

Table 11: Adaptive attacks that know $K$, $\alpha_{\max}$, and $\tau_q$ (Mistral-DNA / TF-M). A scores only non-routed tokens; B subtracts the smoothing-induced loss inflation.

| **Attack** | Standard | Adaptive A | Adaptive B | Adaptive A+B |
|---|---|---|---|---|
| Loss AUC ↓ | 56.48 | 57.62 | 56.83 | 58.58 |
| Shadow AUC ↓ | 52.89 | 53.21 | 53.15 | 54.07 |

Table 12: Wall-clock training time on Mistral-DNA / TF-M. Times for `GenoGuard` exclude the one-time fine-tuning of the reference model.

| **Defense** | **Time/epoch (s)** | **Total (s)** | **Relative** |
|---|---|---|---|
| Clean | 20.1 | 100.5 | 1.00× |
| EMT | 21.0 | 105.2 | 1.05× |
| DP-SGD ($\sigma = 1.0$) | 48.0 | 240.0 | 2.39× |
| `GenoGuard` | 36.1 | 180.5 | 1.80× |

`GenoGuard` adds $1.80\times$ overhead over clean fine-tuning, less than DP-SGD's $2.39\times$, which computes per-sample gradients. The reference model is fine-tuned once on $\mathcal{D}_{\text{ref}}$ before target training; this one-time cost does not scale with target-model epochs.

### A.10 Motif Overlap of High-Risk Spans

To test whether the token hotspots of Section 4 correspond to biologically meaningful sequence, we map token spans back to nucleotide coordinates and compare them with transcription-factor binding motifs from JASPAR 2024 [35]. The analysis uses the trained models only, with no retraining. A span counts as overlapping if it intersects a motif match.

Table 13: Overlap between token spans and known JASPAR transcription-factor motifs. High-risk spans are enriched relative to both controls (Fisher's exact test, odds ratio 7.2, $p < 10^{-4}$).

| **Span set** | **Motif overlap rate (%)** |
|---|---|
| High-risk (top 10% by risk score) | **28.5** |
| Low-risk (bottom 10%) | 5.2 |
| Random spans | 4.9 |

High-risk spans overlap known motifs over five times as often as low-risk or random spans, so the token-level risk score singles out regulatory sequence patterns rather than arbitrary fragments. This supports the region-level attribution maps as a tool for both privacy auditing and biological interpretation.

## 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 our two main claims—that membership leakage in fine-tuned GFMs concentrates on a small fraction of samples and sparse token spans, and that our adaptive defense `GenoGuard` improves privacy with minor utility loss—both of which are supported by Sections 4 and the experiments.

   Guidelines:

   - The answer [N/A] 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 [N/A] 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: Appendix A.1 discusses the absence of a formal privacy guarantee, evaluation on public benchmarks rather than access-controlled patient cohorts, the classification fine-tuning scope, and sensitivity to reference-data shift, together with broader impacts.

   Guidelines:

   - The answer [N/A] 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: [N/A]

Justification: The paper presents an empirical defense and does not include formal theoretical results or proofs.

Guidelines:

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

4. **Experimental result reproducibility**

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

Answer: [Yes]

Justification: The Experiments section and Appendix A.4 describe the models, GUE benchmark tasks, attack configurations (Appendix A.5), defense configurations (Appendix A.6), and hyperparameters needed to reproduce our main results.

Guidelines:

- The answer [N/A] 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 use publicly available models (Mistral-DNA, Nucleotide Transformer, HyenaDNA) and the public GUE benchmark, provide implementation details in Appendix A.4, and release our code.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.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://neurips.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) necessary to understand the results?

   Answer: [Yes]

   Justification: Training/test splits follow the GUE protocol; optimizer, learning rate, batch size, and other hyperparameters are reported in Appendix A.4 and A.6.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: The key comparison against DP-SGD is repeated over three seeds with mean$\pm$std and paired $t$-tests (Table 6). Due to compute cost, the full attack–defense–task grid in the main tables is reported from single runs, averaged across GUE subtasks.

   Guidelines:

   - The answer [N/A] 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: GPU type and training configuration are given in Appendix A.4.

   Guidelines:

   - The answer [N/A] 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. The work uses publicly available genomic benchmarks and pretrained models, contains no personally identifiable information, and aims to improve privacy of fine-tuned models.

   Guidelines:

   - The answer [N/A] 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: Our work is a privacy-preserving defense for genomic foundation models, with positive impact on protecting sensitive genomic data in clinical and biomedical applications; broader impacts are discussed in detail in Appendix A.1.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] 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., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: We do not release any new high-risk models or scraped datasets; we use publicly available genomic foundation models and the public GUE benchmark.

Guidelines:

- The answer [N/A] 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: We cite Mistral-DNA, Nucleotide Transformer, HyenaDNA, the GUE benchmark, and all attack/defense baselines, and use each under its public license.

Guidelines:

- The answer [N/A] 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: `GenoGuard` is documented in Section 4 and Appendix A.4, and the code is publicly released.

    Guidelines:

    - The answer [N/A] 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: [N/A]

    Justification: The paper does not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer [N/A] 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: [N/A]

    Justification: The paper does not involve human subjects, so IRB approval is not applicable.

Guidelines:

- The answer [N/A] 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 rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The core method does not use LLMs as an important or non-standard component; the studied models are genomic foundation models, not general-purpose LLMs.

    Guidelines:

    - The answer [N/A] 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 in the NeurIPS handbook for what should or should not be described.