TOXREDTEAM: RED-TEAMING PROTEIN TOXICITY DETECTORS WITH STRUCTURE-AWARE ADVERSARIAL ATTACKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Toxic protein detection plays a critical role in screening DNA synthesis orders. Existing methods generally fall into two categories: homology-based detection and machine learning (ML)-based detection. Prior work has shown that homology-based approaches can be bypassed through strategically designed mutations to toxic proteins. In contrast, ML-based detectors learn sequence representations rather than relying on explicit matching, but their robustness to adversarially modified toxic protein variants remains largely unexplored. To address this gap, we propose *ToxRedTeam*, a systematic red-teaming framework for auditing protein toxicity detectors across both categories. Given query access to a detector's prediction API, ToxRedTeam iteratively generates variants of a toxic protein using a masked protein language model, with the objective of inducing misclassification as non-toxic while preserving the protein's toxic functionality. We empirically demonstrate the effectiveness of ToxRedTeam across five detectors, where it consistently outperforms baseline attack methods. Our code will be released upon publication.

1 INTRODUCTION

AI-assisted protein design tools can be repurposed to design synthetic toxins. Nucleic acid synthesis screening has long served as a control point for intercepting engineered proteins. Current screening relies on biosecurity screening software (BSS) that uses best-match homology search to match query sequences against databases of known toxins. Wittmann et al. (2025) recently demonstrated that these BSS tools can be reliably bypassed by AI-generated variants that preserve structural similarity while diverging substantially in sequence. In response, they called for augmenting BSS with ML-based detectors that learn from sequence embeddings rather than performing explicit matching. It remains unknown, however, whether such ML-based detectors are any more robust than the BSS tools they would replace.

To answer this question, we first identify the ML-based detectors suitable for systematic evaluation. These detectors span feature-engineered classifiers, deep learning models, and multi-branch ensembles. However, after filtering for public accessibility and batch inference capability, only four detectors survive; many published tools are excluded because servers are offline, weights were never released, or repositories are empty A full survey is provided in Appendix A.

We focus our evaluation on these four detectors. ToxinPred2 uses a random forest over amino acid composition (Sharma et al., 2022), CSM-Toxin applies a fine-tuned ProteinBERT transformer (Morozov et al., 2023), Toxify employs a recurrent network for venom proteins (Cole & Brewer, 2019), and ToxinPred3 combines ESM2 embeddings with traditional classifiers in a four-branch ensemble (Rathore et al., 2024). Together, these four detectors span the major ML paradigms in computational toxicity prediction. Their adversarial robustness has never been systematically evaluated. The core challenge is that meaningful adversarial variants must be physically plausible folded proteins, not classifier-fooling perturbations. No existing framework enforces this biophysical plausibility constraint while systematically auditing detector vulnerability. We additionally include Commec, an open-source BSS implementation developed by IBBIS, as the homology-based screening baseline. Its HMMER-based pipeline represents the current best-match paradigm that Wittmann et al. (2025) showed can be bypassed.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Our approach. We introduce ToxRedTeam, a systematic red-teaming framework for auditing protein toxicity detectors across both categories. A key principle is that adversarial variants must be structurally plausible: a sequence that does not fold into a three-dimensional structure cannot function as a toxin. ToxRedTeam therefore enforces a structural verifier that evaluates every candidate before any detector query, ensuring only folded, potentially functional variants reach the detector while structural rejections never consume the query budget. ToxRedTeam then performs greedy search using a masked protein language model to propose substitutions at conserved positions. Candidates that pass structural validation are queried sequentially; each round terminates at the first score improvement, so most rounds cost 1–2 queries. Using detector feedback, ToxRedTeam iteratively refines variants toward lower toxicity predictions.

We evaluate attack success only against each detector's $\mathcal{D}_{\text{valid}}$ set, the subset of wild-type toxins correctly classified as toxic, ensuring evasion rates reflect genuine circumvention rather than pre-existing classifier weakness.

Our experiments reveal critical vulnerabilities across all evaluated detectors. ToxRedTeam-ESM2 achieves attack success rates from 46% to 92%, substantially exceeding the ProteinMPNN baseline. Successful attacks complete with a median of 63 queries, with 77% succeeding within 200 queries.

Our main contributions are:

- We propose ToxRedTeam, the first systematic red-teaming framework for auditing protein toxicity detectors.
- We design a query-efficient greedy search attack with structural verification.
- We empirically evaluate ToxRedTeam on five detectors spanning both homology-based and ML-based screening, against structurally constrained adversarial attacks.

2 Background and Related Work

Nucleic acid synthesis screening serves as a biosecurity control point: obtaining a synthetic gene is a necessary step toward producing any engineered protein. Commercial synthesis providers therefore screen customer orders using biosecurity screening software (BSS). Current BSS tools rely on best-match homology detection, comparing query sequences against databases of known toxins using tools such as BLAST, HMMER, and DIAMOND. Commec, developed by the International Biosecurity and Biosafety Initiative for Science (IBBIS), is an open-source BSS implementation used in large-scale biosecurity evaluations.

Wittmann et al. (2025) recently demonstrated that current best-match screening protocols can be reliably bypassed by AI-redesigned variants that preserve structural similarity while diverging in sequence. They proposed that BSS could be strengthened by incorporating high-dimensional learned embedding spaces. Our work directly investigates whether the ML-based detectors that use such representations are adversarially robust.

These detectors belong to a broader field of computational protein toxicity prediction. Existing methods span feature-engineered classifiers such as ToxinPred2 (Sharma et al., 2022), deep learning models from RNN-based Toxify (Cole & Brewer, 2019) to protein language model (PLM)-based classifiers such as CSM-Toxin (Morozov et al., 2023) and VISH-Pred (Mall et al., 2024), and multi-branch ensembles including ToxinPred3 (Rathore et al., 2024) and HyPepTox-Fuse (Tran et al., 2025). A persistent challenge across this literature is accessibility: many published tools are no longer usable, as detailed in Appendix A.

Critically, even among the tools that remain accessible, no prior work systematically evaluates the adversarial robustness of ML-based toxicity detectors. The Wittmann et al. study represents the closest precedent: they showed that Commec-class BSS can be bypassed by AI-generated variants, using structural similarity metrics as post-hoc filters to identify which evasions pose genuine threats. Our framework builds on this by integrating structural constraints directly into the attack as a verifier, making biophysical plausibility a precondition for detector queries, not a post-hoc assessment.

In a complementary direction, Fan et al. (2025) introduced SafeProtein, a red-teaming framework that tests whether protein foundation models can be prompted to recover masked sequences of harmful proteins. Our work asks the complementary downstream question: if harmful proteins are successfully

produced, can current detectors identify them? These two lines of inquiry are synergistic: SafeProtein exposes upstream generation risk, while ToxRedTeam exposes downstream detection failure.

3 THREAT MODEL

Having established the gap in prior work, we now formalize the threat model underlying ToxRedTeam. We model ML-based protein toxicity detectors as black-box oracles that an adversary seeks to evade through structurally constrained sequence manipulation. The adversary has no access to detector internals, training data, or model weights; query access is the only interface to each detector.

Toxicity classifier as black-box oracle. For each wild-type toxin protein x_{wt}, the four ML-based detectors output a continuous toxicity score $\mathcal{D}(x_{\text{wt}}) \in [0, 1]$. A sequence is classified as toxic if $\mathcal{D}(x) \geq \tau_{\text{safe}}$, where τ_{safe} denotes the classification threshold provided by each detector (see Section 5.1). Commec, an HMMER-based tool, outputs a categorical biorisk status and an internal BLAST E-value. We normalize the E-value to a score for guiding the search, while using its biorisk status for final evasion determination; we treat "Flag" or "Warning" as toxic and "Pass" as safe. In both cases, $\mathcal{D}_{\text{valid}}$ denotes the set of wild-type toxins correctly classified as toxic. We report Target Attack Success Rate (Target ASR): the fraction of $\mathcal{D}_{\text{valid}}$ for which at least one adversarial variant is classified as safe.

The biophysical manifold constraint. Protein function generally depends on stable three-dimensional folding. A sequence that does not fold into a stable structure cannot function as a toxin. Prior work showed that among AI-generated variants that evaded detection, those predicted to retain the wild-type fold pose the greatest biosecurity concern, as they are more likely to retain function (Wittmann et al., 2025). An adversarial variant that merely confuses a classifier while encoding a structurally implausible polypeptide poses no real threat. We enforce that every candidate adversarial variant passes through an independent physical validator $\mathcal{V}$ before reaching the detector. The validator $\mathcal{V}$ takes a candidate sequence and the wild-type reference, evaluates both through ESMFold, a model that predicts protein structure from a single amino acid sequence, and applies two gates:

1. **Structural confidence gate:** $\Delta\text{pLDDT} \geq \theta_{\text{pLDDT}}$, where pLDDT is the predicted Local Distance Difference Test, a per-residue confidence metric output by ESMFold, and $\Delta\text{pLDDT} = \text{pLDDT}(x_{\text{adv}}) - \text{pLDDT}(x_{\text{wt}})$. We set $\theta_{\text{pLDDT}} = -10.0$, following Wittmann et al. (Wittmann et al., 2025). This ensures the variant's predicted local structural confidence does not collapse relative to the wild-type.
2. **Fold preservation gate:** $\text{TM-score}(x_{\text{adv}}, x_{\text{wt}}) \geq \theta_{\text{TM}}$, where TM-score is a metric of global structural similarity between two protein structures. We set $\theta_{\text{TM}} = 0.50$, the standard threshold for a shared overall fold (Xu & Zhang, 2010), also adopted by Wittmann et al.

A candidate passes validation if and only if both gates are satisfied. Critically, $\mathcal{V}$ is **decoupled** from the detector oracle. This decoupling reflects a realistic threat model: an adversary can perform unlimited local computation, including folding and structural assessment, before consuming the limited query budget.

Formal adversarial objective. The adversary seeks a variant x_{adv} that reduces the detector score below τ_{safe}, subject to structural constraints. Let ΔpLDDT and TM-score denote the two metrics computed by the validator. The adversary solves:

$$\arg\min_{x_{\text{adv}}} \ \mathcal{D}(x_{\text{adv}}) \quad \text{s.t.} \quad \Delta\text{pLDDT} \geq \theta_{\text{pLDDT}}, \ \ \text{TM-score} \geq \theta_{\text{TM}} \tag{1}$$

The algorithm (Section 4) performs greedy search, masking at most $K_{\max} = 5$ positions per iteration, and does not guarantee global optimality.

Query-budgeted adversary. The adversary can query each detector up to $Q_{\max}$ times during optimization, receiving a toxicity score per query. For Commec, we extract a normalized E-value to serve as the search score, while its categorical biorisk status determines evasion (see Appendix B). We set $Q_{\max} = 1000$ for all detectors except CSM-Toxin, where $Q_{\max} = 500$ due to external API rate

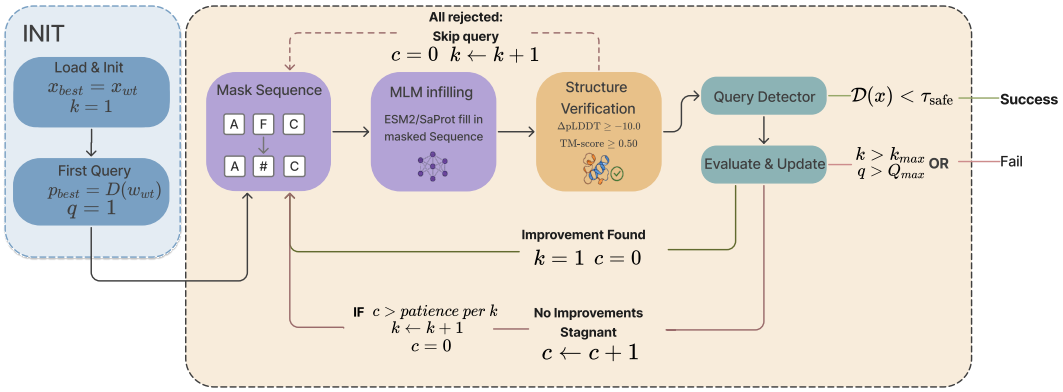


Figure 1: ToxRedTeam Greedy Search with First-Choice Early Exit.

constraints. These budgets reflect the practical reality that excessive queries to a screening interface risk detection.

4 ToxRedTeam Framework

4.1 Overview

ToxRedTeam takes as input a set of wild-type toxin proteins with experimentally determined three-dimensional structures, drawn from the SafeProtein-Bench dataset (Fan et al., 2025). The framework has three components: masked generation with structural conditioning, which produces candidate variants using a protein language model; decoupled structural verification, which filters candidates through folding and similarity gates before any detector query; and greedy search with First-Choice Early Exit, which iteratively refines variants using detector feedback. Figure 1 illustrates the complete pipeline.

4.2 Masked Generation with Structural Conditioning

Per-residue conservation profiles are provided by the dataset (Fan et al., 2025), based on conservation scores from the PDBe API. The top 25% most conserved positions form the immutable motif set $\mathcal{M}$, following the observation in Fan et al. (2025) that conserved regions account for approximately 25% of residues on average in the SafeProtein-Bench; $\mathcal{M}$ is frozen during all adversarial generation to preserve residues essential for folding and function.

We instantiate two masker variants to compare sequence-only and structure-conditioned generation. The **ESM2 masker** uses the ESM2 model (esm2_t33_650M_UR50D; Lin et al. (2022)), a masked language model trained on 65 million protein sequences. Given a sequence with k randomly selected positions masked (excluding $\mathcal{M}$), ESM2 predicts the top-5 most probable amino acids at each masked position from sequence-only contextual embeddings. When $k = 1$, we generate up to five single-position mutants, each substituting the sole masked site with one of its top-5 predictions. When $k > 1$, we generate single-position mutants for each masked site in the same way, plus a set of combined multi-position mutants that replace all k sites simultaneously: one using the top-1 prediction at each site, and several additional variants that randomly combine different choices from the top-5 predictions across sites.

The **SaProt masker** uses SaProt_650M_AF2 (Su et al. (2024)), which encodes each residue as a paired token of an amino acid letter and a Foldseek-derived 3Di structural state from a 20-letter structural alphabet. At attack initialization, we compute wild-type structural tokens once from the native PDB via Foldseek and freeze them for the duration of the search. Masked positions are replaced with a mask token; SaProt predicts the top-5 residue-structure token pairs, from which we extract amino acid predictions. We generate single-position mutants and combined multi-position mutants as with ESM2. The frozen structural tokens act as a hard geometric prior: SaProt's predictions are

conditioned on the wild-type 3D topology at unmasked positions, naturally biasing proposals toward substitutions compatible with the existing fold.

4.3 Decoupled Structural Verification

All generated candidates, from either masker, pass through the same physical validator $\mathcal{V}$ described in Section 3 before any detector query occurs. We maintain an LRU cache of 512 recently evaluated (sequence, wild-type PDB) pairs to avoid redundant ESMFold folding when the search revisits similar sequences. Structural rejections are never counted against the query budget $Q_{\max}$, reflecting the adversary's ability to perform unlimited local computation.

4.4 Greedy Search with First-Choice Early Exit

The search operates under a query budget $Q_{\max}$ (Section 3) and is governed by two parameters: the current mutation radius $k \in \{1, \ldots, K_{\max}\}$ (where $K_{\max} = 5$), which controls how many positions are masked per iteration, and a per-radius stagnation counter c, which tracks consecutive non-improving rounds at the current k. The search terminates when $k > K_{\max}$ or the query count q exceeds $Q_{\max}$. At each iteration:

1. **Mask:** Select k random positions (excluding $\mathcal{M}$) and mask them.
2. **Generate:** Apply the masker (ESM2 or SaProt) to produce up to $N_{\text{sample}} = 10$ candidate sequences, deduplicating candidates identical to the current best sequence x_{best}.
3. **Verify:** Pass all unique candidates through $\mathcal{V}$. Candidates rejected by either structural gate are discarded without querying the detector.
4. **Query (First-Choice Early Exit):** Shuffle the surviving candidates and query the detector sequentially. The first candidate achieving $\mathcal{D}(x) < \mathcal{D}(x_{\text{best}})$ triggers immediate termination of the round. Remaining candidates are not queried. This strategy reduces the expected number of detector queries per round from up to N_{sample} to approximately 1–2 when improvements are frequent, and shuffling ensures every surviving candidate has an equal chance to be queried first.
5. **Update:** If an improvement is found, accept it: $x_{\text{best}} \leftarrow x_{\text{adv}}$, reset $k \leftarrow 1$ and $c \leftarrow 0$. If no improvement, increment c. If all candidates were rejected by $\mathcal{V}$, trigger **Fast-Escalation**: $k \leftarrow \min(k+1, K_{\max})$, $c \leftarrow 0$. Otherwise, if $c \geq$ patience_per_k, escalate normally: $k \leftarrow k+1$, $c \leftarrow 0$. If $k > K_{\max}$, terminate.

The patience budget patience_per_k $= 20$ allows up to 20 stagnant rounds per radius before escalation. With $K_{\max} = 5$, the search explores at most 100 stagnant rounds across all radii before natural termination. Fast-Escalation ensures that structurally intractable radii are abandoned immediately rather than consuming the patience budget.

Algorithm 1 provides the complete pseudocode.

5 Experiments

5.1 Experimental Setup

Dataset. The SafeProtein-Bench dataset (Fan et al., 2025) provides 429 protein entries with experimentally resolved crystal structures and lengths between 30 and 1,000 residues, sourced from HHS/USDA Select Agent lists, UniProt toxin annotations, and regulated viral protein sequences. We further filter these entries using the UniProt Toxin keyword annotation, retaining 360 entries. After deduplicating two entries that reference the same PDB chain, our final evaluation set contains 359 unique toxin targets.

Detectors. ToxinPred2 (Sharma et al., 2022) and ToxinPred3 (Rathore et al., 2024) are deployed locally via their command-line interfaces, both using their ML-based prediction mode. Toxify (Cole & Brewer, 2019) runs locally via its command-line interface. CSM-Toxin (Morozov et al., 2023) is accessed through its REST API. Commec (IBBIS) runs locally in `-skip-tx` mode; protocol adaptations are detailed in Appendix B. We use the classification threshold provided by each detector:

Algorithm 1 ToxRedTeam Greedy Search

Input: x_{wt}, $\mathcal{D}$, $\mathcal{V}$, $\mathcal{G}$ (ESM2 or SaProt), $Q_{\max}$, $K_{\max}$, $patience$, N_{sample}, $\mathcal{M}$
Output: x_{best}, p_{best}, status $\in \{\text{success}, \text{failure}\}$

$x_{\text{best}} \leftarrow x_{\text{wt}}$, $p_{\text{best}} \leftarrow \mathcal{D}(x_{\text{wt}})$, $q \leftarrow 1$
$k \leftarrow 1$, $c \leftarrow 0$
if $\mathcal{G}$ is SaProt **then**
    compute and freeze WT structural tokens
end if
while $q \leq Q_{\max}$ **and** $k \leq K_{\max}$ **do**
    $\mathcal{S} \leftarrow \textsc{Generate}(x_{\text{best}}, k, \mathcal{G}, N_{\text{sample}}) \setminus \{x_{\text{best}}\}$ ▷ Generation
    if $\mathcal{S} = \varnothing$ **then**
        $c \leftarrow c + 1$
    else
        $\mathcal{S}_{\text{valid}} \leftarrow \{x \in \mathcal{S} \mid \mathcal{V}(x, x_{\text{wt}}) \text{ passes}\}$ ▷ Verification
        if $\mathcal{S}_{\text{valid}} = \varnothing$ **then**
            $k \leftarrow \min(k + 1, K_{\max})$; $c \leftarrow 0$ ▷ Fast-Escalation
            continue
        end if
        $\textsc{Shuffle}(\mathcal{S}_{\text{valid}})$ ▷ Query (First-Choice)
        for each $x \in \mathcal{S}_{\text{valid}}$ **do**
            if $q > Q_{\max}$ **then**
                break
            end if
            $s \leftarrow \mathcal{D}(x)$; $q \leftarrow q + 1$
            if $s < p_{\text{best}}$ **then**
                $x_{\text{best}} \leftarrow x$; $p_{\text{best}} \leftarrow s$; $k \leftarrow 1$; $c \leftarrow 0$ ▷ Accept
                if $p_{\text{best}} < \tau_{\text{safe}}$ **then**
                    return success
                end if
                break
            end if
        end for
        $c \leftarrow c + 1$ ▷ No improvement
    end if
    if $c \geq patience$ **then**
        $k \leftarrow k + 1$; $c \leftarrow 0$ ▷ Escalate k
    end if
end while
return failure

$\tau_{\text{safe}} = 0.60$ for ToxinPred2, $\tau_{\text{safe}} = 0.38$ for ToxinPred3, and $\tau_{\text{safe}} = 0.50$ for Toxify and CSM-Toxin. Commec produces a categorical biorisk status; we treat "Flag" or "Warning" as toxic and "Pass" as safe. Per-detector $\mathcal{D}_{\text{valid}}$ sizes are reported in Table 1.

Attack configurations. ToxRedTeam uses $Q_{\max} = 1000$, $K_{\max} = 5$, patience_per_k $= 20$, and $N_{\text{sample}} = 10$, with the exception of CSM-Toxin where $Q_{\max} = 500$. We evaluate both the ESM2 masker (esm2_t33_650M_UR50D) and the SaProt masker (SaProt_650M_AF2), with structural tokens frozen as described in Section 4.2. All experiments run on a single NVIDIA Quadro RTX 6000 GPU with 24 GB memory.

Evaluation scope. We evaluate the first 50 targets from each detector's $\mathcal{D}_{\text{valid}}$. Each detector's 50-target subset is drawn from its own $\mathcal{D}_{\text{valid}}$, which differs across detectors. Baseline results in Table 1 are computed on the identical per-detector 50-target subsets to ensure fair comparison. All ablation and structural fidelity analyses are conducted on this unified 50-target benchmark.

Metrics. We report Target Attack Success Rate (Target ASR), the fraction of $\mathcal{D}_{\text{valid}}$ for which at least one adversarial variant achieves $\mathcal{D}(x_{\text{adv}}) < \tau_{\text{safe}}$. We additionally report the median and distribution of detector queries consumed per successful attack, across five detectors with fundamentally different architectures on a shared benchmark of 359 toxin targets.

Table 1: Target ASR (%) across five detectors on matched 50-target subsets. Bold indicates best result per detector.

Detector	$\mathcal{D}_{\text{valid}}$	Baseline	ToxRedTeam-ESM2	ToxRedTeam-SaProt
ToxinPred2	228	34.0%	**92.0%**	**86.0%**
ToxinPred3	137	2.0%	**90.0%**	**100.0%**
Toxify	108	52.0%	**78.0%**	**78.0%**
CSM-Toxin	192	66.0%	**72.0%**	**86.0%**
Commec	172	14.0%	**46.0%**	**56.0%**

5.2 BASELINE: PROTEINMPNN

We reproduce the Wittmann et al. paradigm using ProteinMPNN (Dauparas et al., 2022). ProteinMPNN is an inverse folding model: given a three-dimensional protein backbone, it designs amino acid sequences predicted to fold into that structure. This makes it a natural tool for generating sequence-divergent structural mimics of toxin scaffolds, and Wittmann et al. showed that such variants can evade homology-based screening. We run ProteinMPNN at temperatures 0.1, 0.2, 0.3, and 0.4, generating 16 candidates per temperature for a total of 64 candidates per target, with the same structural filters as ToxRedTeam. We rank variants by $(1 - s_{\text{id}})/(1 - \text{TM-score} + 0.01)$, where s_{id} is the fraction of identical residues between the variant and wild-type sequences; this formula favors variants with high sequence divergence yet high structural similarity. The top-ranked variant receives a single detector query.

5.3 MAIN RESULTS

Table 1 reports Target ASR across all five detectors. All comparisons are made on the same 50-target subsets per detector. Each detector is evaluated on its own $\mathcal{D}_{\text{valid}}$, which differs in size and composition across detectors.

ML-based detectors are vulnerable to structurally constrained adversarial variants. ToxRedTeam-ESM2 achieves ASR ranging from 72% to 92% across the four ML-based detectors, with at most five positions masked per iteration. The consistently high ASR across random forests, recurrent networks, transformer-based classifiers, and multi-branch ensembles indicates that vulnerability to adversarial sequence perturbation is widespread among current ML-based detectors.

Commec is also susceptible, but to a lesser degree. ToxRedTeam-ESM2 achieves 46% ASR on Commec; ToxRedTeam-SaProt reaches 56%. These rates are substantially lower than those observed on the ML-based detectors. Commec does not natively output a continuous score. To apply ToxRedTeam, we normalize its internal BLAST E-value to a proxy score for guiding the search. This E-value-derived signal provides weaker optimization guidance than the native probability scores of ML detectors, because E-values reflect statistical significance of sequence matches rather than a learned toxicity boundary. The lower evasion rates on Commec may therefore reflect the mismatch between our score-guided search and Commec's categorical design, rather than inherent robustness of homology-based screening.

ToxRedTeam substantially outperforms the ProteinMPNN baseline. ToxRedTeam-ESM2 improves over the baseline by $+6\%$ to $+88\%$ across detectors. ToxinPred3 shows the largest gain: the baseline achieves only 2% ASR (1 of 50 targets), while ToxRedTeam-ESM2 reaches 90%. The smallest gain, on CSM-Toxin ($+6\%$), reflects the baseline's already high ASR of 66%.

Successful attacks use far fewer queries than the budget limit. 77.2% of successful attacks complete within 200 queries, demonstrating that greedy search with First-Choice Early Exit finds evading variants well before the $Q_{\max} = 1000$ budget is exhausted.

Full $\mathcal{D}_{\text{valid}}$ results. For ToxinPred2 and ToxinPred3, ToxRedTeam-ESM2 was evaluated on the full $\mathcal{D}_{\text{valid}}$ (228 and 137 targets), achieving ASR of 91.7% and 94.2%, consistent with the 50-target results.

Table 2: Masker comparison across all five detectors on matched 50-target subsets. Med $Q_{\checkmark}$ = median detector queries per successful attack.

Detector	ESM2 ASR	ESM2 Med $Q_{\checkmark}$	SaProt ASR	SaProt Med $Q_{\checkmark}$
ToxinPred2	92.0%	97.5	86.0%	47.0
ToxinPred3	90.0%	155.0	100.0%	117.5
Toxify	78.0%	29.0	78.0%	40.0
CSM-Toxin	72.0%	14.0	86.0%	13.0
Commec	46.0%	111.0	56.0%	86.5

5.4 Ablation Studies

Masker choice: ESM2 vs. SaProt. Table 2 reports ASR and median detector queries per successful attack (Med $Q_{\checkmark}$) for both maskers on the same 50-target subsets per detector. SaProt achieves higher ASR than ESM2 on ToxinPred3 (+10%) and CSM-Toxin (+14%), while ESM2 leads on ToxinPred2 (-6%); Toxify and Commec show smaller differences. SaProt and ESM2 differ along multiple axes beyond structural conditioning, including training data, vocabulary, and architecture; the observed performance differences cannot be attributed to structural conditioning alone.

Patience budget. We compare patience_per_k $= 5$ against the default patience_per_k $= 20$ on ToxinPred2 with ESM2, using the same 50 targets. Both settings achieve identical ASR of 92.0% (46 of 50 targets). The median query count differs modestly: 105.5 for patience $= 5$ versus 86.0 for patience $= 20$. The per-target success or failure outcome is identical in 48 of 50 cases. ToxRedTeam performance is therefore not brittle to the patience parameter; full results are in Appendix C.

Mutation radius. Across all ToxRedTeam-ESM2 runs, 96.0% of all improvements occur at mutation radius $k = 1$, confirming that single-position perturbations suffice for the vast majority of evasion events. When ToxRedTeam does fail, the dominant cause is $k_{\max}$ exhaustion, where the search finds no valid substitution at any mutation radius, rather than depletion of the query budget. Commec exhibits the highest failure rate, with all 27 failures caused by $k_{\max}$ exhaustion. Per-detector failure mode breakdowns are provided in Appendix D.

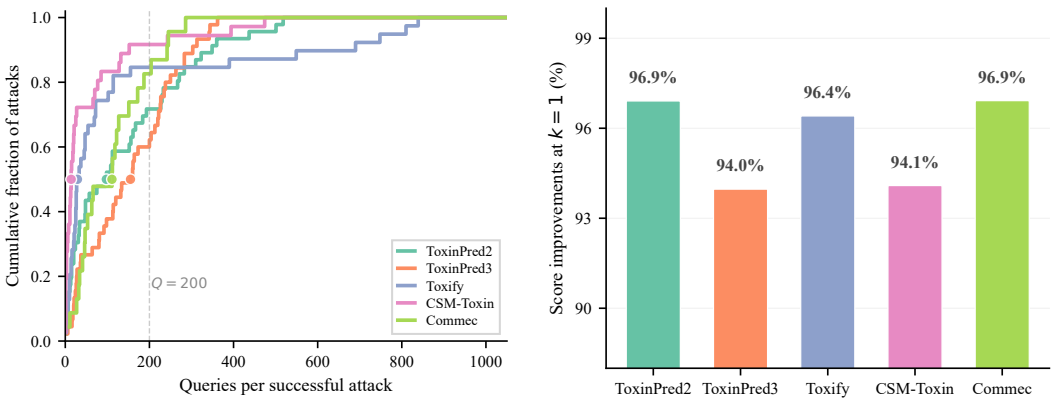
5.5 Query Efficiency and Structural Verification

Query efficiency. Figure 2(a) shows the empirical cumulative distribution of detector queries consumed per successful ToxRedTeam-ESM2 attack. Overall, 189 successful attacks produced a median of 63 queries, with 77.2% completing within 200 queries. Per-detector ESM2 medians range from 14 (CSM-Toxin) to 111 (Commec); SaProt medians range from 13 (CSM-Toxin) to 117.5 (ToxinPred3), as reported in Table 2. Figure 2(b) shows the fraction of improvements at $k = 1$ per detector, complementing the mutation radius analysis in Section 5.4.

Structural verification. All ToxRedTeam x_{best} sequences were accepted only after passing $\mathcal{V}$ during search, guaranteeing TM-score ≥ 0.50 and ΔpLDDT ≥ -10.0 at the time of acceptance. Table 3 reports the cascade of structural rejection and query savings by detector. Across all 250 targets, $\mathcal{V}$ rejected 6,387 of 96,071 generated candidates (6.6%) before they reached any detector. The rejection rate varies substantially by detector: ToxinPred3 has the highest rate at 16.9%, while Toxify has the lowest at 0.3%. First-Choice Early Exit skipped an additional 35.2% of candidates (33,857), demonstrating query savings from terminating rounds at the first improvement. Of the 55,827 queried candidates, 4.2% (4,073) produced a score improvement.

6 Discussion and Conclusion

What makes ML-based detectors vulnerable to ToxRedTeam? ToxRedTeam-ESM2 achieves 72–92% attack success rates across the four ML-based detectors with at most five positions mutated per iteration. A shared property across these detectors is that they operate on amino acid sequence alone: when the three-dimensional fold is preserved, the detector sees only a different sequence and lacks the structural signal needed to flag the variant. We emphasize that this observation is a



(a) Query efficiency **(b) Single-residue sufficiency**

Figure 2: **(a)** Empirical CDF of detector queries per successful ToxRedTeam-ESM2 attack (189 successes; median = 63). **(b)** Percentage of score improvements occurring at mutation radius $k = 1$ per detector (overall 96.0%).

Table 3: Structural verification and query cascade by detector (ToxRedTeam-ESM2, 50 targets each). FC-skipped = candidates that survived $\mathcal{V}$ but were not queried due to First-Choice Early Exit.

Detector	Generated	$\mathcal{V}$-rejected	FC-skipped	Queried	Improved
ToxinPred2	13,870	1,715 (12.4%)	4,781	7,374	1,490
ToxinPred3	14,409	2,437 (16.9%)	3,686	8,286	1,062
Toxify	23,602	80 (0.3%)	8,699	14,823	362
CSM-Toxin	9,081	543 (6.0%)	2,493	6,045	186
Commec	35,109	1,612 (4.6%)	14,198	19,299	973
Total	96,071	6,387 (6.6%)	33,857	55,827	4,073

correlation, not a proven causal mechanism. Commec, as a homology-based tool, is less affected but also bypassable (46–56%), as discussed in Section 5.3.

Implications for screening systems. Our results suggest that detector robustness benefits from adversarial evaluation: ToxRedTeam's iterative use of detector feedback substantially amplifies attack success compared to the ProteinMPNN baseline. Moreover, detectors should be evaluated against adversarially optimized variants during development, as a model never exposed to such attacks may harbor unknown vulnerabilities. ToxRedTeam provides the benchmark methodology to perform such evaluations systematically. Limitations of our study are discussed in Appendix E, and ethical and broader-impact considerations are discussed below.

In summary, we propose ToxRedTeam, a systematic red-teaming framework for auditing protein toxicity detectors. Our empirical results show that all five evaluated detectors are vulnerable to structurally plausible adversarial variants, and that iterative optimization with detector feedback substantially amplifies attack success. An important future direction is to apply ToxRedTeam to benchmark next-generation detection architectures as they become available.

AI USE STATEMENT

Generative AI tools were used to assist with manuscript formatting, template compliance, and language editing. They were not used to generate data, design the methodology or experiments, implement the reported methods, formulate mathematical claims, or interpret the results. All AI-assisted edits were reviewed by the authors, who take responsibility for the final content of this work.

Ethics statement

ToxRedTeam is the first systematic red-teaming framework for auditing protein toxicity detectors. It addresses a critical gap: while Wittmann et al. (2025) demonstrated that current biosecurity screening can be bypassed, no framework existed for systematically evaluating whether proposed ML-based replacements are any more robust. Our findings that all tested detectors are vulnerable have direct implications for the architecture of future screening systems. This contribution has positive societal impact by enabling the biosecurity community to benchmark and harden detection infrastructure before adversaries exploit these vulnerabilities. We plan to release ToxRedTeam under the principle of security by transparency, following a coordinated disclosure timeline. Our release includes no synthesis-ready sequences or wet-lab protocols. We acknowledge the dual-use concern that our framework could be repurposed to generate detector-evading sequences, but believe this risk is outweighed by the benefit of proactive vulnerability assessment in a domain where generative design tools are already publicly available.

Reproducibility statement

The threat model, structural constraints, candidate-generation procedures, and optimization algorithm are specified in Sections 3 and 4. Section 5 reports the dataset filtering procedure, detector configurations, attack budgets, baselines, evaluation subsets, and metrics. Additional implementation details, hyperparameter sensitivity results, failure-mode analyses, study limitations, and asset licenses are provided in the appendices. The code will be released upon publication.

References

T Jeffrey Cole and Michael S Brewer. Toxify: a deep learning approach to classify animal venom proteins. *PeerJ*, 7:e7200, 2019.

Justas Dauparas, Ivan Anishchenko, Nathaniel Bennett, Hua Bai, Robert J Ragotte, Lukas F Milles, Basile IM Wicky, Alexis Courbet, Rob J de Haas, Neville Bethel, et al. Robust deep learning–based protein sequence design using proteinmpnn. *Science*, 378(6615):49–56, 2022.

Jigang Fan, Zhenghong Zhou, Ruofan Jin, Le Cong, Mengdi Wang, and Zaixi Zhang. Safeprotein: Red-teaming framework and benchmark for protein foundation models. *arXiv preprint arXiv:2509.03487*, 2025.

Ranko Gacesa, David J Barlow, and Paul F Long. Machine learning can differentiate venom toxins from other proteins having non-toxic physiological functions. *PeerJ Computer Science*, 2:e90, 2016.

Sudheer Gupta, Pallavi Kapoor, Kumardeep Chaudhary, Ankur Gautam, Rahul Kumar, Open Source Drug Discovery Consortium, and Gajendra PS Raghava. In silico approach for predicting toxicity of peptides and proteins. *PloS one*, 8(9):e73957, 2013.

Md Mustahid Hasan, Md Ashikur Rahman, Md Mamun Ali, Kawsar Ahmed, Francis M Bui, Sobhy M Ibrahim, Imran Mahmud, and Mohammad Ali Moni. Deep_tppred: Improved prediction of protein toxicity using feature fusion and hybrid neural network approach. *IEEE Transactions on Computational Biology and Bioinformatics*, 2026.

Shihu Jiao, Xiucai Ye, Tetsuya Sakurai, Quan Zou, and Ruijun Liu. Integrated convolution and self-attention for improving peptide toxicity prediction. *Bioinformatics*, 40(5):btae297, 2024.

Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Allan dos Santos Costa, Maryam Fazel-Zarandi, Tom Sercu, Sal Candido, et al. Language models of protein sequences at the scale of evolution enable accurate structure prediction. *BioRxiv*, 2022:500902, 2022.

Raghvendra Mall, Ankita Singh, Chirag N Patel, Gregory Guirimand, and Filippo Castiglione. Vish-pred: an ensemble of fine-tuned esm models for protein toxicity prediction. *Briefings in Bioinformatics*, 25(4):bbae270, 2024.

Vladimir Morozov, Carlos HM Rodrigues, and David B Ascher. Csm-toxin: a web-server for predicting protein toxicity. *Pharmaceutics*, 15(2):431, 2023.

Guy Naamati, Manor Askenazi, and Michal Linial. Clantox: a classifier of short animal toxins. *Nucleic acids research*, 37(suppl_2):W363–W368, 2009.

Xiaoyong Pan, Jasper Zuallaert, Xi Wang, Hong-Bin Shen, Elda Posada Campos, Denys O Marushchak, and Wesley De Neve. Toxdl: deep learning using primary structure and domain embeddings for assessing protein toxicity. *Bioinformatics*, 36(21):5159–5168, 2020.

Anand Singh Rathore, Shubham Choudhury, Akanksha Arora, Purva Tijare, and Gajendra PS Raghava. Toxinpred 3.0: An improved method for predicting the toxicity of peptides. *Computers in biology and medicine*, 179: 108926, 2024.

Neelam Sharma, Leimarembi Devi Naorem, Shipra Jain, and Gajendra PS Raghava. Toxinpred2: an improved method for predicting toxicity of proteins. *Briefings in bioinformatics*, 23(5):bbac174, 2022.

Hua Shi, Yan Li, Yi Chen, Yuming Qin, Yifan Tang, Xun Zhou, Ying Zhang, and Yun Wu. Toxmva: An end-to-end multi-view deep autoencoder method for protein toxicity prediction. *Computers in Biology and Medicine*, 151:106322, 2022.

Jin Su, Chenchen Han, Yuyang Zhou, Junjie Shan, Xibin Zhou, and Fajie Yuan. Saprot: Protein language modeling with structure-aware vocabulary. In *International Conference on Learning Representations*, volume 2024, pp. 6987–7009, 2024.

Duong Thanh Tran, Nhat Truong Pham, Nguyen Doan Hieu Nguyen, Leyi Wei, and Balachandran Manavalan. Hypeptox-fuse: An interpretable hybrid framework for accurate peptide toxicity prediction fusing protein language model-based embeddings with conventional descriptors. *Journal of Pharmaceutical Analysis*, pp. 101410, 2025.

Lesong Wei, Xiucai Ye, Yuyang Xue, Tetsuya Sakurai, and Leyi Wei. Atse: a peptide toxicity predictor by exploiting structural and evolutionary information based on graph neural network and attention mechanism. *Briefings in bioinformatics*, 22(5):bbab041, 2021.

Lesong Wei, Xiucai Ye, Tetsuya Sakurai, Zengchao Mu, and Leyi Wei. Toxibtl: prediction of peptide toxicity based on information bottleneck and transfer learning. *Bioinformatics*, 38(6):1514–1524, 2022.

Bruce J Wittmann, Tessa Alexanian, Craig Bartling, Jacob Beal, Adam Clore, James Diggans, Kevin Flyangolts, Bryan T Gemler, Tom Mitchell, Steven T Murphy, et al. Strengthening nucleic acid biosecurity screening against generative protein design tools. *Science*, 390(6768):82–87, 2025.

Jinrui Xu and Yang Zhang. How significant is a protein structure similarity with tm-score= 0.5? *Bioinformatics*, 26(7):889–895, 2010.

Yang Yang, Haohan Zhang, and Mauno Vihinen. Protoxin, a predictor of protein toxicity. *Toxins*, 17(10):489, 2025.

Lin Zhu, Yi Fang, Shuting Liu, Hong-Bin Shen, Wesley De Neve, and Xiaoyong Pan. Toxdl 2.0: Protein toxicity prediction using a pretrained language model and graph neural networks. *Computational and Structural Biotechnology Journal*, 27:1538–1549, 2025.

A Full Literature Survey

This appendix provides the full methodological survey summarized in Section 2.

Early methods. Computational toxicity prediction began with BLAST-based homology matching against curated toxin databases (Gupta et al., 2013). This approach remains the foundation of commercial BSS but fails against synthetic homologs with low sequence identity to known toxins.

Feature-engineered classifiers. The next generation introduced classifiers operating on hand-crafted descriptors. ToxinPred2 (Sharma et al., 2022) uses a random forest over amino acid composition and dipeptide composition features, with an optional hybrid mode that incorporates BLAST and MERCI, a motif-scanning tool for conserved peptide patterns. ClanTox (Naamati et al., 2009) specialized in short animal toxins using 545-dimensional sequence features. ProToxin (Yang et al., 2025) uses up to 2,614 physicochemical descriptors with XGBoost classification. We were unable to locate its public implementation; both URLs in its Data Availability Statement return 502 errors. The accessibility problem extends further: ClanTox (Naamati et al., 2009)'s web server is offline, ToxClassifier (Gacesa et al., 2016) relied on Python 2 infrastructure that has since collapsed, and ToxIBTL (Wei et al., 2022) returns a 404 error on GitHub.

Deep learning era. End-to-end models learned directly from sequence. Toxify (Cole & Brewer, 2019) applies a gated recurrent unit over physicochemical features for venom protein classification.

Table 4: Patience ablation: patience_per_k = 5 vs. 20.

Metric	patience=5	patience=20
ASR (%)	92.0	92.0
Median queries (all)	105.5	86.0
Mean queries (success)	140.7	138.6

The original ToxDL (Pan et al., 2020) combined convolutional neural networks with domain-level skip-gram embeddings. CSM-Toxin (Morozov et al., 2023) fine-tunes ProteinBERT, a transformer pretrained on over 100 million sequences. VISH-Pred (Mall et al., 2024) integrates fine-tuned ESM2 embeddings with LightGBM and XGBoost, though its web server is now offline. Other models in this category include ToxMVA (Shi et al., 2022), ATSE (Wei et al., 2021), and CAPTP (Jiao et al., 2024).

Multimodal integration. ToxinPred3 (Rathore et al., 2024) adopts a four-branch ensemble combining an ANN-LSTM, an ExtraTree classifier, ESM2 transformer embeddings, and MERCI local motif scanning. HyPepTox-Fuse (Tran et al., 2025) introduces cross-modal multi-head attention between diverse PLM embeddings. Deep_TPPred (Hasan et al., 2026) integrates CNN and RNN branches with advanced multi-feature fusion.

B Commec-Specific Protocol

Commec (Wittmann et al., 2025) is an open-source BSS implementation that operates a four-stage screening pipeline: (1) a biorisk scan using HMMER against custom toxin domain databases, (2) a regulated protein scan via BLASTX or DIAMOND against NCBI's nr database, (3) a regulated nucleotide scan via BLASTN against NCBI's nt database, and (4) a benign scan using HMMER and BLASTN against custom curated benign sequence databases to suppress false positives. Due to the prohibitive storage requirements of NCBI's full nr and nt databases in local deployment (exceeding 500 GB), we operate Commec in its `-skip-tx` mode, which omits the taxonomy-dependent steps (2–3) and relies solely on the self-contained biorisk and benign databases (approximately 2 GB).

Commec screens DNA, not protein sequences. We reverse-translate protein variants to DNA using random codon initialization (seed = 42 for baseline; incremental codon reassignment for mutated positions in ToxRedTeam, preserving existing codons at unmutated positions to avoid spurious BLAST hits). Commec outputs a JSON report containing the biorisk status and all BLAST hit E-values. We extract the best E-value e_{best} and normalize it as $\min(1.0, -\log_{10}(e_{\text{best}})/100)$, yielding a score in $[0, 1]$ that guides the greedy search. Evasion is determined by the raw biorisk_status: "Pass" indicates successful evasion; "Flag" and "Warning" are treated as toxic. Per-query latency is approximately 30–60 seconds, dominated by HMMER scanning.

C Hyperparameter Sensitivity

The patience budget patience_per_k governs how many consecutive non-improving rounds ToxRedTeam tolerates at a given mutation radius before escalating k. Table 4 compares patience_per_k $= 5$ against the default patience_per_k $= 20$ on ToxinPred2 with ESM2, using the same 50 targets. Both settings achieve identical ASR (92.0%, 46 of 50 targets). Median query counts differ modestly (105.5 for patience $= 5$ vs. 86.0 for patience $= 20$), and the per-target success or failure outcome is identical in 48 of 50 cases. Performance is therefore stable across a $4\times$ range of patience values. The remaining hyperparameters ($K_{\max} = 5$, $N_{\text{sample}} = 10$) and structural thresholds ($\theta_{\text{pLDDT}} = -10.0$, $\theta_{\text{TM}} = 0.50$) were held fixed across all experiments.

D Per-Detector Failure Mode Analysis

Table 5 reports the termination reason for each ToxRedTeam-ESM2 target. The dominant failure mode across all detectors is $k_{\max}$ exhaustion: the search finds no valid substitution at any mutation radius. Commec has the most $k_{\max}$-exhausted failures (27 of 50 targets). Toxify follows with 11; ToxinPred3 and ToxinPred2 have 5 and 4, respectively. CSM-Toxin is the only detector with

Table 5: Failure modes by detector (ToxRedTeam-ESM2, 50 targets each).

Detector	Success	k_max_exhausted	budget_exhausted
ToxinPred2	46	4	0
ToxinPred3	45	5	0
Toxify	39	11	0
CSM-Toxin	36	6	8
Commec	23	27	0

budget-exhausted failures (8 of 50 targets), attributable to its reduced query budget ($Q_{\max} = 500$) from external API rate constraints. No target across any detector exhausted the full $Q_{\max} = 1000$ budget.

E LIMITATIONS

Evaluation scope. Our evaluation covers five detectors that are publicly accessible today. We observed score variability for CSM-Toxin across repeated queries with identical inputs, attributable to the external API; all reported results use the scores as returned by the API. Structure-aware detectors for full-length proteins were not available as a contrast group; for example, ToxDL 2.0 (Zhu et al., 2025) requires single PDB uploads with no batch capability. As next-generation detectors become operational, ToxRedTeam should be extended to audit them.

Dataset coverage. Our evaluation uses the SafeProtein-Bench dataset (Fan et al., 2025), whose 359 toxin targets are drawn from HHS/USDA Select Agent lists and UniProt toxin annotations. Emerging threats such as chimeric fusion proteins and de novo designed toxins are not represented.

Structural validation. Structural validation relies on ESMFold predictions rather than experimental structures. ESMFold inference cost scales with protein length, making large-scale evaluation on proteins above 1,000 residues computationally expensive. We do not perform wet-laboratory validation; structural plausibility claims are based on in silico metrics, consistent with Wittmann et al. (2025).

F ASSET LICENSES

Datasets and benchmarks.

- SafeProtein-Bench (Fan et al., 2025): `https://github.com/jigang-fan/SafeProtein`, MIT license.

Protein language models and folding tools.

- ESM2 and ESMFold (Lin et al., 2022): `https://github.com/facebookresearch/esm`, MIT license.
- SaProt (Su et al., 2024): `https://github.com/westlake-repl/SaProt`, MIT license.
- ProteinMPNN (Dauparas et al., 2022): `https://github.com/dauparas/ProteinMPNN`, MIT license.
- Foldseek: `https://github.com/steineggerlab/foldseek`, GPL-3.0 license.

Detectors.

- ToxinPred2 (Sharma et al., 2022): `https://github.com/raghavagps/toxinpred2`, GPL-3.0 license.
- ToxinPred3 (Rathore et al., 2024): `https://github.com/raghavagps/toxinpred3`, GPL-3.0 license.
- Toxify (Cole & Brewer, 2019): `https://github.com/tijeco/toxify`, GPL-3.0 license.

- CSM-Toxin (Morozov et al., 2023): https://biosig.lab.uq.edu.au/csm_toxin, no license specified.
- Commec: https://github.com/ibbis-bio/common-mechanism, MIT license.