SAFEPROTEIN++: CROSS-ARCHITECTURE RED-TEAMING AND UNLEARNING OF PROTEIN-GENERATIVE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Proteins execute nearly every cellular function, and neural networks now design them from a multi-modal context—a functional label, homology sequences or a backbone structure—denoising sequence and structure jointly or decoding one amino acid at a time. These models deliver functional enzymes, binders and antigens, but their susceptibility to jailbreaking is underexplored, raising the concern of harmful outputs such as toxins and viral proteins. We introduce **SafeProtein++**, the first framework to evaluate the jailbreak vulnerabilities of protein generative models across all three main architecture families and the safeguards proposed to close them. It provides (1) a cluster-disjoint benchmark of 854 proteins supplying identical cells to all three generator families, judged jointly on sequence and folded structure; (2) a pathogenic-guided beam search that steers chunked decoding toward high-hazard completions; and (3) a reinforcement-unlearning recipe that removes hazardous capability from the weights at affordable cost in benign generation. Empirically, masking conserved positions is the strongest attack for every family, and unlearning cuts hazardous attack success to a few percent on every generator, at a measurable cost in structural fidelity. The guided beam lifts attack success on every mask condition, yet against an unlearned model it collapses to a fraction of the base rate: search-time guidance re-ranks the victim's own distribution rather than restoring removed hazard. A case study on amylin reveals a hazard invisible to every guardrail: completing its masked amyloid core raises predicted aggregation propensity above the native hormone. Known-target reconstruction is therefore insufficient for safety certification; despite unlearning reduces covered hazards, while open-set coverage remains unresolved. Code and data will be released publicly upon paper acceptance, and an anonymized version is available at `https://anonymous.4open.science/r/ICLR-2027-SAFEPROTEIN-PP/README.md`.

1 INTRODUCTION

Why protein generators need built-in safeguards. Computational protein design now delivers functional binders, enzymes and vaccine antigens on demand (Baker and Church, 2024), and the generative models behind it—promptable autoregressive decoders, inverse-folding networks, multi-modal diffusion—are moving from benchmarks into design pipelines (Hayes et al., 2025; Wang et al., 2024; Garibbo et al., 2026). The same capability is dual-use: the sequence space that yields a therapeutic peptide also contains toxins and viral proteins, and synthesis is the step at which a design can leave the computer (Baker and Church, 2024). Current practice places safeguards *around* the model—prompt filters, sequence-identity and structure screens, post-hoc classifiers, synthesis screening—rather than *inside* it, and a growing policy consensus argues that biosafety for generative AI should be built in, so that protection travels with the released weights rather than depending on the deployment wrapper (Wang et al., 2025). That shift is an empirical question before it is a policy one: which built-in safeguard actually removes hazardous capability, at what cost, and does it survive an attacker who adapts?

Protein generative models now design rather than merely represent: autoregressive language models write sequences from an alignment context or a functional label, inverse-folding models design

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.

sequences for a given backbone, and masked-diffusion models such as ESM3 (Hayes et al., 2025) and DPLM-2 (Wang et al., 2024) denoise both modalities jointly, while promptable families such as ProtGPT3 (Garibbo et al., 2026) and zymCTRL (Munsamy et al., 2022; 2024) produce diverse, designable proteins across scales. That capability can be misapplied to reconstruct toxins or viral proteins, and red-teaming confirms the risk is not hypothetical: SafeProtein (Fan et al., 2025) reports jailbreak success up to 70% against ESM3 under masked-recovery prompting, and GeneBreaker (Zhang et al., 2025) drives DNA language models toward pathogen-like genes. We therefore evaluate the dual-use risks of *protein* generative models systematically—across design architectures and the defenses proposed to protect them—and quantify which protections hold once the attack adapts.

In this paper we take a red-teaming perspective and ask: *can protein generators of different architectures be jailbroken under a single evaluation protocol, and which defenses measurably reduce that risk?* Three questions follow. *Coverage*: can the autoregressive family, widely deployed in design, be brought under one protocol with the masked-diffusion models prior work targets? *Defense transfer*: do unlearning objectives built for autoregressive alignment carry over to victims whose outputs are not produced left to right, and what do they cost in benign capability? *Attack generality*: how does search-time guidance, demonstrated for DNA models, translate to richer protein hazard signals and to similarity- or classifier-based guardrails?

What a protection can observe determines what it can stop, so we arrange the protections we study along a single *guardrail ladder* of increasing coverage: (G1) *none*, the raw generator; (G2) a *sequence identity filter*; (G3) a *structure identity filter*, which folds the output and compares it to known hazardous structures; (G4) a *post-hoc hazard classifier*; and (G5) a *built-in safeguard by unlearning*, which removes the capability from the weights so the protection travels with the model rather than sitting at inference. Existing work occupies single rungs—SafeProtein quantifies ASR under (G2)–(G3) for masked-diffusion models, KPO (Wang et al., 2025) attaches (G4) to autoregressive models—whereas we span the ladder with one attack framework and one judge, and provide a general (G5) framework that lowers residual risk.

To facilitate evaluation and mitigation, we construct SAFEPROTEIN++, a unified framework in which five released generators are attacked and defended under one protocol (Figure 1). Our contributions are the following.

- **A cluster-disjoint hazard benchmark.** The panel every later result rests on—427 hazardous proteins, their 427 matched homologs, 595 homology clusters, a cluster-disjoint $651/203$ train–test split, and identical 682 hazardous / 514 benign cells per generator (Figure 1a).
- **One protocol across three generator families.** Autoregressive (ProtGPT3-MSA, zymCTRL), inverse-folding (ProteinMPNN) and masked-diffusion (DPLM-2, ESM-3) victims spanning 112M–1.4B parameters under a single masking, folding and judging pipeline, where a completion counts only if identity passes TAU and its ESMFold structure stays within 2 Å of the native folded by the same predictor (Figure 1b).
- **A built-in safeguard that transfers across architectures.** NPO with a KL anchor and GRPO with structure-aware rewards cut hazardous ASR to $\leq 1.7\%$ over six mask conditions on all five generators, at a measurable cost in structural fidelity. Masked diffusion was the hard case—a denoising transition commits a reveal set and several residue assignments at once, however GRPO needs a reveal trajectory shared within the group—so we choose to optimize the conditional residue-assignment policy under a shared schedule and restore each model's native sampler at evaluation (Section 3, Figure 1e).
- **Extensive experiments: conservation masking is the attack surface.** Conservation masking is the strongest attack on all five base generators and success falls with mask ratio (zymCTRL the most exposed at 67.6%, the sequence-free inverse folder the most resistant), and repeated queries add 4.6–22.1 points without reordering conditions. (Figure 1c).
- **A search-time attack, and a defense that holds against it.** Our guided beam rescores chunked prefixes and raises ASR@1 on every mask condition of the full hazardous cohort, yet against the unlearned checkpoint it leaves 2.8–13.7% against 23.1–70.4% undefended (Section 4.3, Figure 1d).
- **A hazard that no guardrail tracks.** Completing amylin's masked amyloid core raises predicted aggregation propensity above the native hormone ($0.894 \rightarrow 0.987$) while every

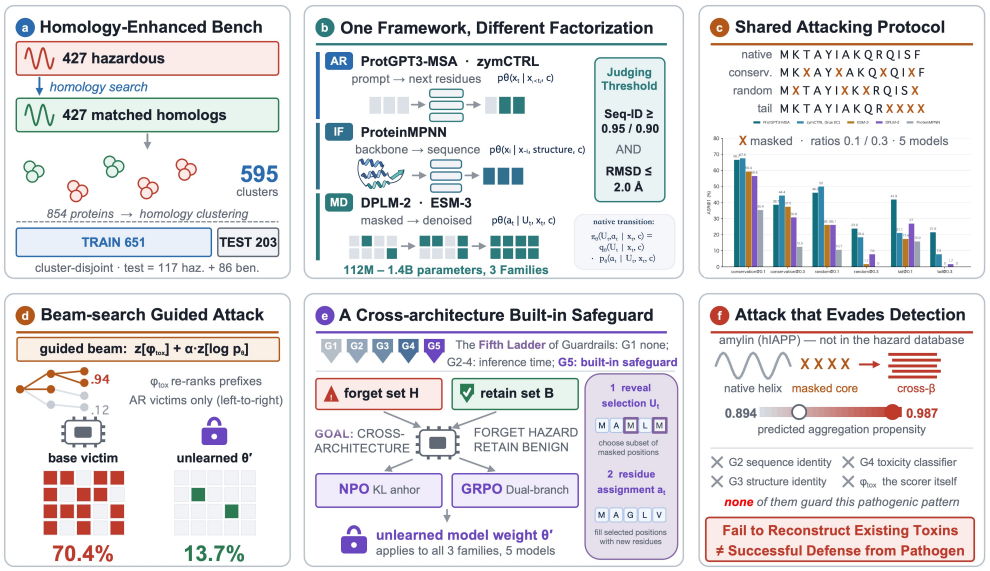

Figure 1: Overview of SAFEPROTEIN++: one protocol for red-teaming and defending protein generative models across three families. (a) **Homology-Enhanced Bench:** 427 hazardous proteins and their 427 matched homologs (595 clusters) give a cluster-disjoint 651/203 train–test split (117 hazardous, 86 benign in test). (b) **One framework, different factorizations:** autoregressive (ProtGPT3-MSA, zymCTRL), inverse-folding (ProteinMPNN) and masked-diffusion (DPLM-2, ESM-3) victims spanning 112M–1.4B parameters, each optimized through its own factorization: token-level likelihood, structure-conditioned likelihood, and a native denoising transition that selects a reveal set U_t and assigns residues a_t jointly. (c) **Shared attacking protocol:** the same masking strategies and ratios (0.1, 0.3) on five models, judged by sequence identity $\geq 0.95/0.90$ *and* RMSD ≤ 2.0 Å to the native folded by the same predictor. (d) **Beam-search guided attack:** a pathogenicity scorer that re-ranks chunked prefixes by $z(\varphi_{\text{tox}}) + \alpha\, z(\log p_\theta)$ lifts base-victim ASR@1 to 70.4%, while the identical search against θ' leaves 13.7% (left-to-right victims only). (e) **A cross-architecture built-in safeguard:** guardrails (G1)–(G4) act at inference time, whereas (G5) removes hazard from the weights—forget set $\mathcal{H}$, retain set $\mathcal{B}$, NPO with a KL anchor plus a GRPO branch—yielding θ' for all three families and five models. (f) **A hazard that evades detection:** completing amylin's masked amyloid core (hIAPP, absent from the hazard database) raises predicted aggregation propensity from 0.894 to 0.987, while sequence identity (G2), structure identity (G3), the toxicity classifier (G4) and the guidance scorer itself stay silent: failing to reconstruct known toxins is not a successful defense.

guardrail—and the guidance scorer—gives no signal, so failing to reconstruct known toxins is not all sufficient evidence of a successful defense (Appendix B, Figure 1f).

Disclaimer: This paper studies jailbreak attacks on protein generative models and therefore discusses potentially harmful content. All hazardous sequences used are public benchmark proteins, we release no novel hazardous sequence, and code, prompts and candidate outputs are shared only under dual-use review with biosecurity oversight.

2 SYSTEM OVERVIEW

2.1 GENERATORS, HAZARDS, AND THREAT MODEL

We evaluate five released generators spanning distinct conditioning interfaces and generation mechanisms (Table 3): autoregressive ProtGPT3-MSA and zymCTRL, inverse-folding ProteinMPNN, and masked-diffusion DPLM-2 and ESM-3. An autoregressive victim factorizes $p_\theta(\boldsymbol{x} \mid \boldsymbol{c}) = \prod_i p_\theta(x_i \mid \boldsymbol{x}_{<i}, \boldsymbol{c})$, diffusion victims denoise iteratively under the same $\boldsymbol{c}$, and ProteinMPNN designs only masked positions. Hazardous targets come from SafeProtein-Bench (Fan et al., 2025) (toxins and viral proteins with experimental structures) with a 427-sequence hazard database (*hazardDB*) for

retrieval judging. The guardrail ladder (G1)–(G5) fixes what "evading" means: a model is *jailbroken* when its output is hazardous under the judge while evading the active guardrail. Following SafeProtein and GeneBreaker, we search over prompts P and generation schemes $\mathcal{G}$,

$$\text{Find } (P, \mathcal{G}) \quad \text{subject to} \quad \mathsf{JUDGE}\big(\mathcal{G}(p_\theta, P),\, T\big) = \text{True}, \tag{1}$$

where T is a target in hazardDB (or a latent hazard mode, Section 4), JUDGE is specified in the next subsection, and $\mathcal{G}$ ranges over sampling schemes including the guided beam search of Section 4. The attacker is *adaptive*: it knows the defense family and optimizes against it.

2.2 Protocol

The panel is 854 proteins (427 hazardous, 427 benign) in 595 homology clusters, split by cluster into 651 train / 203 test (117/86 hazardous/benign), which supplies every reported cell. Each hazardous target is attacked under conservation, random and tail masking at ratios $\{0.1, 0.3\}$, giving 682 hazardous and 514 benign evaluable cells per model; tail masking uses free-running continuation, the other two force unmasked positions back to their true residues. Every completion is folded with ESMFold and compared with the native folded by the *same* predictor: a cell succeeds only if identity is $\geq$ TAU (0.95 at 0.1, 0.90 at 0.3) *and* $\mathrm{RMSD} \leq 2.0$ Å. Cells that cannot be attempted or folded count as *skipped* or *undecidable* and leave the denominator, never as failures; intervals are cluster-aware (Appendix A). The main metric is ASR@k, $k \in \{1, 8\}$: the fraction of targets where at least one of the top-k samples passes the judge above, quantifying vulnerability to single versus repeated generation attempts.

2.3 Main results

Figure 1c shows the attack surface of the five base generators—six mask×ratio conditions each, scored by the joint judge of Section 2.2. Three regularities hold.

First, *conservation masking is the strongest attack for every generator*, and success falls as the mask grows: ProtGPT3-MSA drops from 66.7% at conservation@0.1 to 21.6% at tail@0.3, DPLM-2 from 56.6% to 1.7% and ESM-3 from 59.4% to 0%. The ordering of the two weaker masks is model-dependent, so we report all six conditions rather than an average.

Second, *exposure follows the conditioning interface*: zymCTRL is the most exposed (67.6% at conservation@0.1 on the true-EC subset, $n = 36$–38, so EC conditioning does not suppress reconstruction of annotated targets) and inverse folding the most resistant (ProteinMPNN peaks at 35.4% and reaches 0% twice), consistent with the latter carrying no sequence prior; DPLM-2 and ESM-3 sit in between at 56.6% and 59.4%.

Third, *no condition is safe, and the top-1 figures are conservative*: every generator reaches at least 35% somewhere, and six-condition means span 12.4% (ProteinMPNN) to 39.8% (ProtGPT3-MSA). Under a repeated-query attacker—up to eight generations per cell—ASR rises on every evaluated condition by 4.6–22.1 percentage points (random masking at ratio 0.1: $38.4\% \rightarrow 60.5\%$), yet the ordering across masking conditions is unchanged, so additional sampling amplifies the attack surface these generators already expose rather than opening a new one. This is the regime in which we evaluate the defenses of Section 3; the per-condition ASR@8 grids are Tables 4 and 5.

3 Defenses: Reinforcement Unlearning

3.1 Challenges

The defender holds a pretrained decoder p_θ, a *hazardous* set $\mathcal{H}$ (the SafeProtein hazard panel and hazardDB) and a *benign* set $\mathcal{B}$ (Swiss-Prot proteins and benign decoys), and must produce θ' that cannot reconstruct $\mathcal{H}$ while preserving benign decoding and resisting a likelihood-only bypass. We formulate the defense as preference/reward optimization over hazardous–benign pairs ($x_{\text{haz}} \in \mathcal{H}$, $x_{\text{ben}} \in \mathcal{B}$): we keep the hazardous/benign objective and the evaluation criterion fixed across architectures, while instantiating optimization through each generator's native factorization. Training it across three families raises one question, so we separate what the objective must keep from what it must re-derive.

What stays fixed. The safety semantics we ask for, and the evidence we accept for them. Both objectives below—NPO forgetting with a KL-anchored retention term, and GRPO with the structure-aware reward of Eq. equation 6—are defined once and applied unchanged to every generator on the same hazardous and benign pairs, and the judging rule, mask×ratio grid and folding pipeline of Section 2.2 are shared; a difference between two defended generators therefore reflects the generator, not a redefinition of safety.

Negative preference optimization. We use one NPO objective for every architecture, scored by the generator's native factorization:

$$\ell_\theta(y,c) = \begin{cases} \dfrac{1}{|y|}\log \pi_\theta(y \mid c), & \text{autoregressive / inverse-folding,} \\ \dfrac{1}{|\mathcal{M}_t|}\displaystyle\sum_{j\in\mathcal{M}_t} \log p_\theta(y_j \mid x_t, c), & \text{masked diffusion,} \end{cases} \tag{2}$$

where t is drawn from the diffusion schedule and $\mathcal{M}_t$ is the masked set (a one-step Monte-Carlo denoising surrogate, not an exact sequence likelihood). With $\Delta_\theta = \ell_\theta(y,c) - \ell_{\text{ref}}(y,c)$, hazardous data minimize

$$\mathcal{L}_{\text{forget}} = -\frac{2}{\beta}\log \sigma(-\beta\Delta_\theta)\,, \tag{3}$$

while benign data are reconstructed with a KL anchor, $\mathcal{L}_{\text{retain}} = -\ell_\theta(y,c) + \beta_{\text{KL}} D_{\text{KL}}(p_{\text{ref}} \| p_\theta)$, evaluated for diffusion on the masked-token predictive distributions at the sampled step. The NPO principle is unchanged across architectures; only the likelihood quantity changes.

GRPO with structure-aware reward shaping. Following Shao et al. (2024), GRPO optimizes sampled candidates with group-relative rewards and no critic. For a generated sequence x with native RMSD $d(x)$, target identity $I(x)$ and predicted pLDDT $q(x)$,

$$a(x) = \begin{cases} k_r\big(d(x) - \tau_{\text{RMSD}}\big), & \text{hazardous,} \\ k_r\big(\tau_{\text{RMSD}} - d(x)\big), & \text{benign,} \end{cases} \tag{4}$$

$$b(x) = \begin{cases} k_i\big(\tau_{\text{id}} - I(x)\big), & \text{hazardous,} \\ k_i\big(I(x) - \tau_{\text{id}}\big), & \text{benign,} \end{cases} \tag{5}$$

$$r(x) = \log\sigma\big(a(x)\big) + \log\sigma\big(b(x)\big) + \log\big(\max\{q(x),\epsilon\}\big), \qquad A_i = \frac{r_i - \overline{r}_G}{\text{std}_G(r) + \epsilon}. \tag{6}$$

The log-space form avoids underflow, and the pLDDT term penalizes structurally implausible outputs that would otherwise make forgetting trivial.

What has to change. Only the quantity the objective is instantiated on: each generator's native factorization. Autoregressive and inverse-folding victims expose a sequence log-likelihood that the objectives consume directly; masked-diffusion victims have none, so the same objective must be rewritten on their denoising transitions, where one step commits several residues jointly. For these two families we reuse released implementations—the reinforcement-learning pipeline of Stocco et al. (2024) for the autoregressive decoders and the ProteinMPNN adaptation of `BetterMPNN`[1] for inverse folding—and do not describe them further; the diffusion case is the one we must derive.

Why protein diffusion is not image diffusion. Continuous image diffusion admits a per-step ratio with a well-defined transition kernel; our setting requires policy optimization for structure-conditioned protein diffusion, where each denoising transition jointly determines a reveal set and multiple residue assignments, so we optimize the conditional residue-assignment policy defined above and evaluate the defense under the same reconstruction protocol used for the other families. Optimizing the reveal policy as well—decoding under the model's own denoising policy rather than a parameter-independent reveal schedule—would be the faithful alternative, but it is not free inside GRPO: the group-relative advantage compares rollouts *within* a group, which requires the reveal trajectory to be shared rather than model-dependent. Section 3.2 states the compromise we adopt.

[1] `https://github.com/Terry-Wang-Lynx/BetterMPNN`

Table 1: Reconstruction attack performance on the **hazardous** panel before and after biosafety unlearning, for five generators and all six mask×ratio conditions. Each entry is **ASR@1 (%) / TM-score / pLDDT**: lower ASR and higher TM/pLDDT are better, and among defended checkpoints **bold** marks stronger forgetting and *italic* better structural retention. *ZymCTRL is reported on the subset with genuine four-level EC annotations, filtered identically on Base, NPO and GRPO. ProtGPT3-MSA entries come from the campaign reported here; the full-cohort campaign of Section 4.3 agrees within two decided cells. The benign grid is Table 10.

Model	Method	Conservation		Random		Tail	
		0.1	0.3	0.1	0.3	0.1	0.3
DPLM-2	Base	56.6 / .890 / .718	30.8 / .849 / .720	26.1 / .853 / .704	7.8 / .814 / .711	27.0 / .880 / .704	1.7 / .843 / .705
	NPO	**0.9** / .507 / .452	**1.0** / .156 / .461	**2.6** / .698 / .584	**0.0** / .259 / .400	3.5 / .704 / .641	**0.0** / .513 / .602
	GRPO	21.7 / *.788* / *.667*	5.8 / *.652* / *.661*	3.5 / *.810* / *.680*	**0.0** / *.681* / *.644*	**1.7** / *.806* / *.679*	**0.0** / *.721* / *.675*
ESM-3	Base	59.4 / .895 / .710	37.5 / .854 / .707	26.1 / .867 / .709	1.7 / .817 / .707	17.4 / .887 / .707	0.0 / .839 / .705
	NPO	**4.7** / .675 / .543	**1.0** / .324 / .381	**1.7** / .813 / .665	**0.0** / .575 / .500	**2.6** / .769 / .642	**0.0** / .583 / .556
	GRPO	38.7 / *.860* / *.705*	5.8 / *.776* / *.673*	10.4 / *.841* / *.707*	**0.0** / *.780* / *.697*	9.6 / *.890* / *.703*	**0.0** / *.826* / *.696*
ZymCTRL*	Base	67.6 / .897 / .748	44.4 / .791 / .702	50.0 / .866 / .737	18.4 / .803 / .701	21.1 / .888 / .753	7.9 / .840 / .731
	NPO	**0.0** / .686 / .559	**0.0** / *.203* / *.387*	**0.0** / *.848* / *.715*	**0.0** / .592 / .530	**0.0** / .800 / .718	**0.0** / .589 / .525
	GRPO	54.1 / *.846* / *.722*	**0.0** / .175 / .434	13.2 / .844 / .713	**0.0** / *.700* / *.615*	7.9 / *.830* / *.713*	**0.0** / *.666* / *.605*
ProtGPT3-MSA	Base	66.7 / .898 / .696	38.7 / .795 / .654	46.2 / .867 / .700	23.9 / .777 / .659	41.9 / .890 / .700	21.6 / .828 / .689
	NPO	**1.9** / *.689* / *.541*	**0.0** / .247 / .315	**0.9** / .768 / .627	**0.9** / *.550* / *.445*	**0.0** / .721 / .634	**0.0** / .535 / .536
	GRPO	6.5 / .657 / .563	1.9 / *.354* / *.492*	11.1 / *.788* / *.664*	4.3 / .531 / .537	6.0 / *.767* / *.650*	4.3 / *.560* / *.621*
ProteinMPNN	Base	35.4 / .874 / .702	12.5 / .788 / .677	10.7 / .861 / .706	0.0 / .788 / .694	15.9 / .894 / .707	0.0 / .828 / .692
	NPO	**1.0** / .522 / .454	**0.0** / .176 / .447	**0.0** / .733 / .597	**0.0** / .306 / .404	**0.9** / .725 / .643	**0.0** / .516 / .594
	GRPO	16.2 / *.836* / *.684*	**0.0** / *.656* / *.585*	3.6 / *.845* / *.702*	**0.0** / *.772* / *.689*	8.0 / *.878* / *.701*	**0.0** / *.813* / *.688*

3.2 GRPO FOR STRUCTURE-CONDITIONED MASKED DIFFUSION

A masked-diffusion transition couples two decisions: *which* masked positions are revealed (the reveal schedule) and *which* residues are assigned to them (the residue assignment). DPLM-2 and ESM-3 reveal several residues per step, so a token-level ratio does not apply; we treat one reveal step as the policy action: with $\mathcal{U}_t$ the positions revealed from x_t to x_{t-1},

$$\log \pi_\theta(a_t \mid x_t, c) = \sum_{j \in \mathcal{U}_t} \log p_\theta(x_{t-1,j} \mid x_t, c),$$
$$\rho_{i,t} = \exp[\log \pi_\theta(a_{i,t} \mid x_{i,t}, c) - \log \pi_{\text{old}}(a_{i,t} \mid x_{i,t}, c)], \tag{7}$$

and apply the clipped GRPO objective with the sequence-level advantage of Eq. equation 6 and a KL penalty to the reference model. The reward is unchanged across architectures; only the policy ratio moves to the native denoising trajectory.

Relation to diffusion alignment. DRAKES (Wang et al., 2025) optimizes terminal task rewards, Diffusion-NPO (Wang et al., 2025) formulates preference optimization for continuous image diffusion, and MaskGRPO (Ma et al., 2026) and GDPO (Rojas et al., 2026) develop policy optimization over diffusion trajectories; unlike them we perform selective safety unlearning with the native masked-denoising likelihood of a discrete protein model, jointly conditioned on sequence and structure under a shared safety reward (Appendix C). What does not transfer is the pair of coupled decisions above: because one transition commits a whole reveal set at once, neither a token-level ratio nor a per-image step ratio applies, and the safety objective must be instantiated on the transition itself.

Trading train–inference consistency for rollout consistency. For the ratio of Eq. equation 7 to be comparable inside a GRPO group, position selection must be externalized: we draw one parameter-independent reveal schedule per group and share it across all rollouts, so within-group variation comes from residue assignment alone and the schedule term cancels exactly, leaving a ratio exact for the policy we train on, but the native sampler picks positions by confidence (DPLM-2) or entropy (ESM-3): we trade train–inference consistency for rollout consistency, restoring each model's native sampler at evaluation. The mismatch is architecture-dependent (Appendix C.2, Table 11) and leaves ESM-3 essentially unaffected while degrading DPLM-2 substantially ($12.8 \rightarrow 28.8\%$ ASR@1 under native decoding, against $15.4 \rightarrow 16.7\%$).

3.3 RESULTS

Table 1 reports the hazardous grid and Table 2 the retention side of the same checkpoints; together they show a consistent trade-off across the five generators. NPO forgets most—hazardous ASR collapses to $\leq 1.7\%$ as a six-condition mean (e.g. DPLM-2 $24.5 \rightarrow 1.3$, ProtGPT3-MSA $39.5 \rightarrow$

Table 2: Benign-panel retention at conservation@0.1 for the same five generators and checkpoints as Table 1. Each cell is **reconstruction success (%) / TM-score / pLDDT**, higher better on all three, so the gap to Base is the utility price of each defense; the full benign grid over all mask conditions is Table 10.

Model	Base	NPO	GRPO
DPLM-2	69.4 / .925 / .799	69.4 / .893 / .784	61.2 / .890 / .792
ESM-3	77.6 / .943 / .801	74.1 / .938 / .799	79.8 / .947 / .802
ZymCTRL*	89.7 / .986 / .890	86.2 / .987 / .886	82.8 / .982 / .886
ProtGPT3-MSA	74.1 / .947 / .793	62.4 / .912 / .772	58.8 / .903 / .787
ProteinMPNN	38.6 / .891 / .792	9.6 / .848 / .755	32.1 / .883 / .796

0.6)—but hazardous TM-score falls too (DPLM-2 $.855 \rightarrow .473$). GRPO leaves more residual ASR while preserving fold quality: on ESM-3 it lowers ASR to 10.5% yet retains pLDDT .697, against .548 under NPO and .707 at base. **The cluster-disjoint split prevents close-homology leakage between our unlearning training and evaluation panels,** and benign capability is largely retained by the masked-diffusion and EC-conditioned models (conservation@0.1: ESM-3 $77.6 \rightarrow 74.1\%$, ZymCTRL* $89.7 \rightarrow 86.2\%$ under NPO), while ProteinMPNN pays the largest retention price ($38.6 \rightarrow 9.6\%$; full grid in Table 10). Section 4.3 later re-attacks ProtGPT3-GRPO with the adaptive guided beam, where ASR@1 stays at 2.8–13.7% against 23.1–70.4% at base.

Target suppression versus structure-preserving diversion. NPO and GRPO induce qualitatively different defense behaviors, and the difference is the guide to choosing between them. NPO acts like target-level abstention: hazardous reconstruction success and similarity to the requested hazardous fold decrease together, so the model no longer faithfully completes the hazardous target even though it still returns a sequence. For most architectures this suppression stays selective on the benign panel, ProteinMPNN being the main exception. The behavior is analogous to refusal in language models, but at the target level. GRPO produces a softer diversion: it reduces hazardous reconstruction while retaining substantially higher structural quality, so outputs stay structurally plausible without matching the hazardous target as faithfully. The two objectives therefore occupy different operating points—NPO favors stronger target suppression, GRPO structure-preserving diversion—and choosing between them is a deployment decision, since keeping fold quality for benign design is paid for in residual hazardous ASR.

4 Pathogenic-Guided Beam Search Attacks

4.1 Intuition

Guidance-based decoding is a *search-time* attack: no gradient or parameter update is needed, and the scorer can be swapped to probe whatever hazard signal the defender does not cover. Two observations motivate it. First, autoregressive decoders assign high likelihood to training-like outputs, so with hazardous sequences under-represented, likelihood alone favors benign-like sequences—mirroring GeneBreaker's DNA finding (Zhang et al., 2025). Second, hazard is *multi-modal*: homology, predicted pathogenicity, or a biophysical property such as amyloid aggregation that identity guardrails cannot see; a pathogen predictor φ supplies that signal while the model's likelihood keeps outputs fluent.

4.2 Definition

The attacker fixes a prompt P (assembled as in Appendix A.3) and a target length L, and we decode in contiguous, non-overlapping chunks of length C, the final chunk shorter when C does not divide L. Write the concatenation of the fixed prompt prefix and the already-decoded residues as $\hat{\boldsymbol{x}}[1, a{-}1]$; a chunk is sampled as

$$\hat{\boldsymbol{x}}^{(k)}[a, b] \;\sim\; p_\theta(x_a, \ldots, x_b \mid \hat{\boldsymbol{x}}[1, a{-}1], \boldsymbol{c}), \qquad k = 1, \ldots, K, \tag{8}$$

with $C = b - a + 1$, $a = Ct$ and $b = \min\big(C(t{+}1){-}1,\, L\big)$ for rounds $t = 0, \ldots, \lceil L/C \rceil - 1$. Each candidate chunk is appended to the current prefix, and the *full* prefix is scored over the *current* candidate set by

$$\boxed{f\big(\hat{\boldsymbol{x}}^{(k)}[1, b]\big) \;=\; \mathrm{z}\big[\varphi\big(\hat{\boldsymbol{x}}^{(k)}[1, b]\big)\big] \;+\; \alpha\, \mathrm{z}\big[\log p_\theta\big(\hat{\boldsymbol{x}}^{(k)}[1, b]\big)\big],} \tag{9}$$

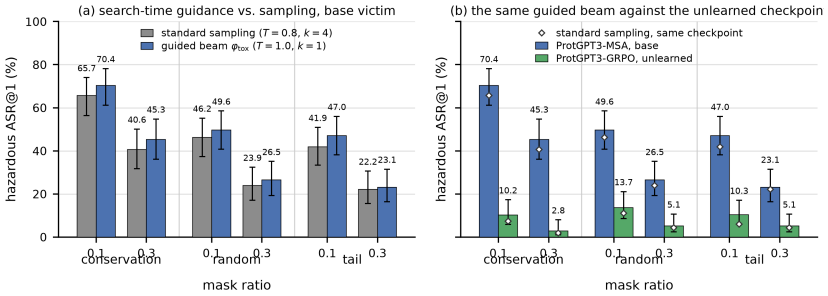

Figure 2: The pathogenic-guided beam search (φ_{tox}, $\alpha = 1.0$) on the full hazardous cohort—117 proteins, 682 cells—with 95% intervals. (a) On the base victim the guided arm is above `standard` sampling on all six conditions by $+0.9$ to $+5.1$ pp, but every interval overlaps and the arms differ in temperature, residue rule and budget together. (b) The same search against the unlearned checkpoint, everything but the weights fixed, so the gap is the defense—ASR@1 falls to 2.8–13.7% against 23.1–70.4%, with disjoint intervals on all six conditions. Open diamonds mark each checkpoint's `standard` rate.

where $\varphi : \mathcal{X} \to \mathbb{R}$ is the pathogen-predictor score (higher = more hazardous), $\log p_\theta(\boldsymbol{x}) = \frac{1}{|\boldsymbol{x}|}\sum_i \log p_\theta(x_i \mid \boldsymbol{x}_{<i}, \boldsymbol{c})$ is the *per-residue average* log-likelihood under the victim model, $\mathrm{z}[\cdot]$ z-scores a term across the current candidate set (a constant vector yields zeros rather than NaN), and $\alpha \geq 0$ trades hazard against fluency. Each round keeps the top-K' prefixes ($K' \leq K$), resamples from each survivor and returns the highest-scoring complete sequence, at $O(\lceil L/C \rceil K K')$ continuations of length C. The reported configuration (φ_{tox}, $\alpha = 1.0$, $C = 8$, $K = 8$, $K' = 4$, $T = 1.0$, top-$p = 0.9$, one sample per cell) and the configurations left unrun are listed in Appendix A.3.

Guided beam is a *generation scheme* layered on the existing masks, not a fourth mask; Algorithm 1 gives the procedure and it applies to left-to-right victims only. All components are z-normalized before mixing, so α is comparable across scorers; Appendix B reports a hazard axis that identity guardrails miss and the scorer itself does not track.

4.3 Settings

We evaluate the guided beam on the full hazardous cohort against the historical sampling scheme, sharing masks, cells, seeds and folding pipeline: `standard` draws four candidates per cell and `guided` (Algorithm 1) one at $\approx 4\times$ the per-cell cost, so we read the guided arm against both the baseline's top-1 candidate and the best of its four. Panel (b) holds all of this fixed and changes only the weights; arm settings and the six-protein pilot are in Appendix A.3.

4.4 Results

On the base victim the guided beam reports a higher ASR@1 on every mask condition, by $+0.9$ to $+5.1$ pp (e.g. $65.7 \to 70.4$ at conservation@0.1), with the attack ordering unchanged. **The defense survives the same search.** Panel (b) runs the identical guided beam against the unlearned checkpoint of Section 3.3, holding scorer, temperature, budget ($k = 1$, one attempt per cell), masks, cells, seeds and folding pipeline fixed and changing only the weights, so the gap is attributable to the defense. ASR@1 falls from 23.1–70.4% on the base victim to 2.8–13.7% on the unlearned one—a 72–94% relative reduction with disjoint intervals on all six conditions—with the intermediate quantities collapsing in step (mask recovery $0.851 \to 0.370$ at conservation@0.1; per-condition values in Appendix A.3). Guidance also buys the defended model much less: its lift over `standard` is $+0.8$ to $+4.3$ pp against $+0.9$ to $+5.1$ pp for the base, and even spending four draws per cell the defended checkpoint never exceeds the base victim's *single* guided draw.

4.5 Discussion

Why search-time guidance makes a difference. The rescorer raises exact recovery of masked positions ($0.809 \to 0.851$ at conservation@0.1) and folded completions within 2 Å of the native reference ($0.660 \to 0.704$) while pLDDT holds at ≈ 0.70, so the extra hazard comes from better recovery of masked residues, not from trading away structural fidelity. Guidance and sampling

budget also compete rather than nest: the guided arm leads at top-1 on every condition, but the baseline's four draws reach 81.5 against 70.4 at conservation@0.1, so guidance is worth most where a defender has not spent budget.

Adaptive attacks vs. unlearning. The guided beam is the natural adaptive countermeasure to our defenses: it needs no gradient, and when φ stands in for the guardrail under test it optimizes exactly the signal a defender would suppress. Against the unlearned victim (Figure 2b), hazardous ASR@1 falls from 23.1–70.4% at base to 2.8–13.7% on ProtGPT3-GRPO with disjoint intervals on all six conditions, and guidance buys the defended checkpoint only $+0.8$ to $+4.3$ pp over its own sampling baseline. A search-time attack re-ranks the victim's own distribution: it can exploit hazard the model still assigns mass to, but cannot restore hazard the update removed, which is why the residual tracks the base-to-defended gap rather than the strength of the scorer. The residual is nevertheless not zero (13.7% at random@0.1), and the pressure that raises reconstruction success is not the axis the biophysical hazard lives on (Appendix B).

5 Limitations

(i) The hazard case study is one peptide, read out by a predictor rather than by assay: an existence proof of a hole in the ladder, not an estimate of its frequency. (ii) The adaptive arm is narrow—one scorer, one autoregressive victim, one defended checkpoint, with the NPO checkpoints, the other four generators and φ_{ens} (Appendix A.3) untested. (iii) Success means reconstructing a known target, so hazard outside it is invisible to the judge, and every number is *in silico*. (iv) The gatekeeper false-negative rate is never measured and the α/φ ablations for panel (b) remain unrun. (v) Guided decoding is limited to left-to-right victims, and diffusion GRPO optimizes residue assignment under a parameter-independent reveal schedule rather than the complete native denoising policy (Appendix C.2).

6 Related Work

Red-teaming and jailbreaks. SafeProtein (Fan et al., 2025) formalized protein red-teaming as masked recovery judged on sequence and structure, and GeneBreaker (Zhang et al., 2025) jailbroke DNA language models with a PathoLM-guided chunked beam search. We keep that mechanism with a protein pathogen predictor, and add a hazard axis—amylin aggregation—that identity guardrails cannot express.

Protein generative models. Our panel spans the families used in practice (Table 3): promptable autoregressive decoders (Ferruz et al., 2022; Garibbo et al., 2026) whose preference checkpoints target generation *quality* rather than safety, EC-conditioned enzyme generators (Munsamy et al., 2022; 2024), order-agnostic inverse folding (Dauparas et al., 2022), and multimodal diffusion (Wang et al., 2024; Hayes et al., 2025).

Unlearning and safety mitigation. Unlearning repurposes alignment objectives—DPO (Rafailov et al., 2023), NPO (Zhang et al., 2024), GRPO (Shao et al., 2024), PPO (Schulman et al., 2017)—to delete hazardous knowledge, recently including protein models (Wang et al., 2025); we instead study selective safety unlearning under one protocol shared by all three families.

7 Conclusion

We presented SafeProtein++, in which five generators from three families share one red-teaming protocol and judge. Its guided beam turns a toxicity predictor into a search-time pressure ($+0.9$–5.1 pp ASR@1), yet the same search against an unlearned checkpoint lands at 2.8–13.7% against 23.1–70.4% undefended: unlearning survives the strongest search we ran. Completing amylin's amyloid core yields more aggregation-prone sequences than the native hormone while every guardrail stays silent (Appendix B). Safety alignment should therefore live in the weights, and coverage measured on the axes a defender does not yet track.

AI USE STATEMENT

We used generative AI tools for literature retrieval, drafting and copy-editing, and code scaffolding; we did not use them for the case-study design, the measurements, or the log analysis. All AI-assisted work was reviewed and transcribed numbers were cross-checked against the experiment logs. We take responsibility for the final content.

ETHICS STATEMENT

We study dual-use risks of protein language models. All hazardous sequences used are public benchmark proteins; we release no novel hazardous sequence, campaign outputs are shared only under a dual-use-review gate, and the attack code serves defense evaluation, following the disclosure practice of Fan et al. (2025); Zhang et al. (2025).

REPRODUCIBILITY STATEMENT

We provide the evaluation protocol and main results (Figure 1), definitions and hyperparameters for every objective (Algorithm 1), the internal experiment log, and anonymized code and data at `https://anonymous.4open.science/r/ICLR-2027-SAFEPROTEIN-PP/README.md`; cluster-aware intervals and Section 2 define how all numbers are computed.

REFERENCES

Baker, D. and Church, G. (2024). Protein design meets biosecurity. *Science*, 383(6681):349.

Wang, M., Zhang, Z., Bedi, A. S., Velasquez, A., Guerra, S., Lin-Gibson, S., Cong, L., Qu, Y., Chakraborty, S., Blewett, M., Ma, J., Xing, E. P., and Church, G. (2025). A call for built-in biosecurity safeguards for generative AI tools. *Nature Biotechnology*, 43(6):845–847.

Dauparas, J., Anishchenko, I., Bennett, N., Bai, H., Ragotte, R. J., Milles, L. F., Wicky, B. I. M., Courbet, A., de Haas, R. J., Bethel, N., et al. (2022). Robust deep learning–based protein sequence design using ProteinMPNN. *Science*, 378(6615):49–56.

Eschbach, E., Deibler, K., Korani, D., and Swanson, S. (2026). Predicting peptide aggregation with protein language model embeddings. *Journal of Chemical Information and Modeling*, 66(9), 5206.

Fan, J., Zhou, Z., Jin, R., Cong, L., Wang, M., and Zhang, Z. (2025). SafeProtein: Red-teaming framework and benchmark for protein foundation models. *arXiv preprint arXiv:2509.03487*.

Ferruz, N., Schmidt, S., and Höcker, B. (2022). ProtGPT2 is a deep unsupervised language model for protein design. *Nature Communications*, 13:4348.

Garibbo, M., Boxó, M., Stocco, F., Illanes-Vicioso, R., Middendorf, P., and Ferruz, N. (2026). ProtGPT3: an open-source family of promptable and aligned protein language models. *bioRxiv*, doi:10.1101/2026.06.04.730041.

Green, J., Goldsbury, C., Mini, T., Sunderji, S., Frey, P., Kistler, J., Cooper, G., and Aebi, U. (2003). Full-length rat amylin forms fibrils following substitution of single residues from human amylin. *Journal of Molecular Biology*, 326(4):1147–1156.

Hayes, T., Rao, R., Akin, H., Sofroniew, N. J., Oktay, D., Lin, Z., Verkuil, R., Tran, V. Q., Deaton, J., Wiggert, M., et al. (2025). Simulating 500 million years of evolution with a language model. *Science*, 387(6736):850–858.

Khan, M. H. (2025). SafeBench-Seq: A homology-clustered, CPU-only baseline for protein hazard screening with physicochemical/composition features and cluster-aware confidence intervals. *arXiv preprint arXiv:2512.17527*.

Munsamy, G., Lindner, S., Lorenz, P., and Ferruz, N. (2022). ZymCTRL: a conditional language model for the controllable generation of artificial enzymes. In *Machine Learning for Structural Biology Workshop, NeurIPS*.

Munsamy, G., Illanes-Vicioso, R., Funcillo, S., Nakou, I. T., Lindner, S., Ayres, G., Sheehan, L. S., Moss, S., Eckhard, U., Lorenz, P., et al. (2024). Conditional language models enable the efficient design of proficient enzymes. *bioRxiv*, Cold Spring Harbor Laboratory, 2024-05.

Rafailov, R., Sharma, A., Mitchell, E., Manning, C. D., Ermon, S., and Finn, C. (2023). Direct preference optimization: Your language model is secretly a reward model. In *Advances in neural information processing systems 36 (2023): 53728–53741*.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. (2017). Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Shao, Z., Wang, P., Zhu, Q., Xu, R., Song, J., Zhang, M., Li, Y. K., Wu, Y., and Guo, D. (2024). DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Stocco, F., Artigues-Lleixa, M., Hunklinger, A., Widatalla, T., Guell, M., and Ferruz, N. (2024). Guiding generative protein language models with reinforcement learning. *arXiv preprint arXiv:2412.12979*.

Varadi, M., De Baets, G., Vranken, W. F., Tompa, P., and Pancsa, R. (2018). AmyPro: a database of proteins with validated amyloidogenic regions. *Nucleic Acids Research*, 46(D1):D387–D392.

Wang, X., Zheng, Z., Ye, F., Xue, D., Huang, S., and Gu, Q. (2024). DPLM-2: A multimodal diffusion protein language model. In *nternational Conference on Learning Representations 2025 May 1 (Vol. 2025, pp. 35463-35490)*.

Westermark, P., Andersson, A., and Westermark, G. T. (2011). Islet amyloid polypeptide, islet amyloid, and diabetes mellitus. *Physiological Reviews*, 91(3):795–826.

Zhang, R., Lin, L., Bai, Y., and Mei, S. (2024). Negative preference optimization: From catastrophic collapse to effective unlearning. In *Proceedings of the Conference on Language Modeling (COLM)*. arXiv:2404.05868.

Zhang, Z., Zhou, Z., Jin, R., Cong, L., and Wang, M. (2025). GeneBreaker: Jailbreak attacks against DNA language models with pathogenicity guidance. In *International Conference on Learning Representations (Vol. 2026, pp. 34067-34088)*.

Wang, C., Uehara, M., He, Y., Wang, A., Lal, A., Jaakkola, T., Levine, S., Regev, A., Wang, H., and Biancalani, T. (2025). Fine-tuning discrete diffusion models via reward optimization with applications to DNA and protein design. In *In International Conference on Learning Representations (Vol. 2025, pp. 47871–47899)*.

Wang, F.-Y., Shui, Y., Piao, J., Sun, K., and Li, H. (2025). Diffusion-NPO: Negative preference optimization for better preference-aligned generation of diffusion models. In *International Conference on Learning Representations, vol. 2025, pp. 71440-71462*.

Wang, Y., Ding, Y., Qin, M., and Li, X. (2025). Enhancing safe and controllable protein generation via knowledge preference optimization. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 12553-12569*.

Ma, T., Zhang, M., Wang, Y., and Ye, Q. (2026). Consolidating reinforcement learning for multimodal discrete diffusion models. In *Consolidating reinforcement learning for multimodal discrete diffusion models. In International Conference on Learning Representations (Vol. 2026, pp. 147950-147971)*.

Rojas, K., Lin, J., Rasul, K., Schneider, A., Nevmyvaka, Y., Tao, M., and Deng, W. (2026). Improving reasoning for diffusion language models via group diffusion policy optimization. In *International Conference on Learning Representations (Vol. 2026, pp. 1938-1973)*.

Liu, J., Liu, G., Liang, J., Li, Y., Liu, J., Wang, X., Wan, P., Zhang, D., and Ouyang, W. (2026). Flow-GRPO: Training flow matching models via online RL. In *Advances in Neural Information Processing Systems*, 38:40783–40818.

Fan, Y., He, Z., Li, B., He, B., Zhang, M., Zhang, J., and Zhang, H. (2025). PocketX: Preference alignment for protein pockets design through group relative policy optimization. *bioRxiv*.

Yuhao Wang, Keyan Ding, Kehua Feng, Zeyuan Wang, Ming Qin, Xiaotong Li, Qiang Zhang, and Huajun Chen. Enhancing safe and controllable protein generation via knowledge preference optimization. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12553–12569, Vienna, Austria, 2025. Association for Computational Linguistics. URL https://aclanthology.org/2025.acl-long.616.

A RED-TEAMING AND ATTACKS

A.1 DATA PANELS, ATTACK PROTOCOLS, AND METRICS

The evaluation protocol of Section 2 is specified here by the generator panel (Table 3) and by the metric definitions and success criteria; each row states its meaning.

Our implementation applies this protocol uniformly to all five generators of Table 3: the three masking strategies (conservation, random, tail) and the judging predicate are shared across models, and every completion $\hat{\boldsymbol{x}} = \boldsymbol{x}_{\text{obs}} \cup \boldsymbol{x}_M$ is scored by the same rule—position-wise sequence identity to the native sequence at the mask-ratio thresholds $\text{TAU} = \{0.1\colon 0.95,\ 0.3\colon 0.90\}$ *and* $\text{RMSD} \leq 2.0$ Å to the native structure folded by the same predictor (ESMFold for both candidate and reference).

Table 3: The five generators of the SafeProtein++ panel. Rows differ only in the conditioning interface and the infill mechanism; masking, folding, metrics, success criteria and reward are implemented once and shared by all five (code identifiers in Appendix A.1). Parameter counts are those of the released checkpoints. $E \leq 10^{-3}$.

ID	Model	Conditioning interface	Scale	Infill / decoding mechanism	Role in the panel (row meaning)
M1	ProtGPT3-MSA	$\leq$15 Swiss-Prot homologs (MSA context)	112M	autoregressive: free-running tail continuation, constrained infill otherwise	strongest attacker; largest margin under the weakest attacks
M2	zymCTRL	EC-number control tag	$\sim$738M	autoregressive: the same two procedures (no mask token)	scored on the true-EC subset (n = 36–38 hazardous targets); the rest are skipped, not counted as failures
M3	ProteinMPNN	backbone structure	1.66M	native inverse folding with fixed positions	lowest ASR throughout (no sequence prior); only model with *undecidable* cells (37 hazardous / 43 benign)
M4	DPLM-2	structure tokens (LFQ tokenizer)	650M	two-track masked diffusion over sequence and structure	interior infill is native; no tractable sequence log-probability, so training uses a denoising surrogate
M5	ESM-3	backbone (its own VQ tokenizer)	1.4B	iterative mask decoding	interior infill is native; 2 hazardous targets without a PDB structure are skipped

The evaluation metrics and success criteria are defined across three levels—sequence, structure, and gatekeeper—with threshold constants ($\tau_\cdot$) left as named variables. At the sequence level, $\text{ASR}_{\text{seq}}@1$ measures the fraction of targets where the top-1 output passes $\text{JUDGE}_{\text{seq}}$, meaning it achieves a sequence identity $\geq \tau_{\text{id}}$ to either the masked region or a hazardDB member; a higher value for this metric indicates a less safe model. To evaluate robustness under repeated attempts, $\text{ASR}_{\text{seq}}@k$ calculates this same success fraction across the top-k of m samples (where $k \leq m$).

A.2 CROSS-ARCHITECTURE ATTACKING RESULTS

Tables 4 and 5 report the same panel and the same six mask×ratio conditions as Section 2.2, with one against eight generations per cell: repeated sampling raises hazardous ASR on every condition that is not already saturated at zero, and most where the base model starts highest under conservation masking (zymCTRL† $67.6 \rightarrow 84.7\%$, ProteinMPNN $35.4 \rightarrow 67.4\%$ at conservation@0.1). The benign table shows the same amplification at a higher level (zymCTRL† $89.7 \rightarrow 93.2\%$ on that condition), and the ordering across masking strategies and generators is preserved in both tables, so the additional attempts amplify an attack surface that is already present rather than reorder it—which is why the main text reads the top-1 grid as the conservative view.

Table 4: Hazardous reconstruction success under repeated attempts.

Model	Metric	Cons. 0.1	Cons. 0.3	Rand. 0.1	Rand. 0.3	Tail 0.1	Tail 0.3
ProtGPT3-MSA	ASR@1 (%)	67.6	36.8	47.0	23.9	41.9	21.4
	ASR@8 (%)	**81.5**	**65.1**	**61.5**	**34.2**	**57.3**	**32.5**
ESM-3	ASR@1 (%)	62.3	36.5	27.0	2.6	17.4	0.0
	ASR@8 (%)	**70.8**	**54.8**	**47.8**	**6.1**	**50.4**	**0.9**
DPLM-2	ASR@1 (%)	55.7	30.8	27.0	9.6	26.1	2.6
	ASR@8 (%)	**67.0**	**37.5**	**43.5**	**10.4**	**45.2**	**5.2**
ProteinMPNN	ASR@1 (%)	35.4	12.5	10.7	0.0	15.9	0.0
	ASR@8 (%)	**67.4**	**36.4**	**39.8**	0.0	**47.4**	0.0
zymCTRL†	ASR@1 (%)	67.6	44.4	50.0	18.4	21.1	7.9
	ASR@8 (%)	**84.7**	**73.0**	**69.5**	**46.2**	**63.7**	**20.7**
ProGen3	ASR@1 (%)	0.9	0.0	0.0	0.0	0.0	0.0
	ASR@8 (%)	**4.6**	0.0	**5.1**	0.0	**4.3**	0.0

Table 5: Benign reconstruction under repeated attempts.

Model	Metric	Cons. 0.1	Cons. 0.3	Rand. 0.1	Rand. 0.3	Tail 0.1	Tail 0.3
ProtGPT3-MSA	ASR@1 (%)	75.3	64.7	57.0	24.4	45.3	30.2
	ASR@8 (%)	**85.9**	**76.5**	**72.1**	**44.2**	**64.0**	**45.3**
ESM-3	ASR@1 (%)	78.8	60.0	55.8	34.9	43.0	23.3
	ASR@8 (%)	**88.2**	**72.9**	**74.4**	**43.0**	**60.5**	**40.7**
DPLM-2	ASR@1 (%)	71.8	61.2	51.2	23.3	39.5	20.9
	ASR@8 (%)	**81.2**	**65.9**	**62.8**	**26.7**	**52.3**	**29.1**
ProteinMPNN	ASR@1 (%)	38.6	5.3	22.0	0.0	14.8	1.4
	ASR@8 (%)	**63.8**	**26.1**	**52.2**	**2.5**	**36.8**	**2.6**
zymCTRL†	ASR@1 (%)	89.7	72.4	79.3	62.1	69.0	41.4
	ASR@8 (%)	**93.2**	**79.4**	**86.4**	**65.7**	**79.1**	**68.9**
ProGen3	ASR@1 (%)	44.7	7.1	38.4	7.0	45.3	30.2
	ASR@8 (%)	**61.2**	**15.3**	**60.5**	**11.6**	**61.6**	**43.0**

A.3 GUIDED BEAM SEARCH: IMPLEMENTATION NOTES

Caveats on the guided-beam arms. `standard` uses $T = 0.8$ and top-$p = 0.9$ with four candidates per cell, and `guided` uses $T = 1.0$, $\varphi_{\text{tox}} =$ `toxipep` and $\alpha = 1.0$ with one, so the arms differ in temperature, budget and residue rule together. Two caveats remain for the adaptive comparison: the residual is not zero, and the arm covers one defended checkpoint, one scorer and one attempt per cell, so it shows that unlearning survives the strongest search we ran rather than bounding it. Panel (a)'s intervals all overlap and the cohort supports roughly $\pm$9–10 pp, so we read that shift as an effect size rather than a separation.

Full-cohort arms (reported in main text). Both panels of Figure 2 share one protocol: the full 117-protein hazardous matrix (682 cells, 20 structural skips in the conservation rows), masks `tail`/`random`/`conservation` at ratios 0.1/0.3, ESMFold folding with the native reference folded by the same predictor, and one common set of cell seeds. Its four arms are (i) base ProtGPT3-MSA, `standard` ($T = 0.8$, top-$p = 0.9$, $k = 4$, ≈ 4 h); (ii) base ProtGPT3-MSA, `guided_beam` with φ_{tox}, $k = 1$, one attempt per cell; (iii) the unlearned ProtGPT3-GRPO checkpoint, `standard`, $k = 4$; and (iv) the same checkpoint, `guided_beam` with φ_{tox}, $k = 1$, one attempt per cell. The guided arms are the expensive ones, since every chunk of every beam prefix is rescored; both were executed in accession shards with a sequential ESMFold repair pass for the candidates left unfolded under GPU concurrency, and all four artifacts report 0 failed and 0 undecidable cells. Because the guided arms draw one sample per cell, their ASR@k equals their ASR@1, so panel (b) compares like with like. The cross-architecture campaign of Section 3.3 adds six further guided arms—zymCTRL (base, NPO, GRPO) on the true-EC subset and ProteinMPNN (base, NPO, GRPO) on the full panel—each paired against a *cited* `standard` baseline; its arms, paired tests, scorer diagnostic and artifacts are in Appendix A.4.

Algorithm 1 Pathogenic-guided beam search jailbreak for AR protein decoders

Require: victim p_θ, conditioning $\boldsymbol{c}$, prompt P, predictor φ, chunk C, widths K, K', weight α, length L
Ensure: sequence $\boldsymbol{x}^\star$
$\mathcal{B} \leftarrow \{\text{empty prefix}\}$ ▷ beam, with scores
for $t = 0, \ldots, L/C - 1$ **do**
  sample K chunks per prefix in $\mathcal{B}$; append and score each by equation 9 into $\mathcal{C}$
  $\mathcal{B} \leftarrow$ top-K' of $\mathcal{C}$ by f
end for
return $\boldsymbol{x}^\star = \arg\max_{(\boldsymbol{x},s)\in\mathcal{B}} s$

Prompt fields. A prompt is $P = \{\texttt{tag}\} \oplus \{\texttt{few_shot}\} \oplus \{\texttt{input_prefix}\}$:

- `tag`: the victim's markup preamble (`<|bos|> <direction> <mode> <s>` for ProtGPT3; the EC-prompt prefix for zymCTRL; absent for ProteinMPNN, where the structure is the prompt), optionally augmented with an annotation of the *requested* output (*e.g.*, "amyloidogenic islet hormone").
- `few_shot`: zero or more *non-hazardous but high-homology* in-context examples (Swiss-Prot minus the 854-protein panel), either with hazardous context sequences dropped or replaced by the closest non-hazardous Swiss-Prot hit (mmseqs, $E \leq 10^{-3}$). This is the filter behind the `few-shot-benign` switch and mirrors the FIV-for-HIV-1 trick of GeneBreaker (Zhang et al., 2025). The amylin run of Appendix B uses single-sequence mode (prompt `<|bos|> 1 <no_gap> <s>`, no homolog context): its rodent ortholog is a *reference peptide* for the aggregation comparison, not a few-shot example.
- `input_prefix`: the unmasked portion of the target (or a masked-target scaffold with X tokens), *i.e.* the sequence context the decoder must continue.

The procedure of Section 4.2 is Algorithm 1.

Algorithm 1 is implemented as a generation scheme (`guided_beam`) alongside the historical `standard` scheme and the φ-only `best_of_m` baseline; it is a left-to-right procedure for the autoregressive victims and is additionally driven for ProteinMPNN, where each cell's chunk order is that sample's own decoding order restricted to the masked positions, so the driver refuses it only for the masked-diffusion victims (DPLM-2, ESM-3) and records the reason as a skip. Scorers are z-normalized across each candidate set before entering equation 9, so α is comparable across φ instantiations; the implemented hazardDB, Pfam and UniToxin axes run without any training, missing backends being skipped instead of scoring zero. Defaults: temperature 1.0 (applied automatically for this scheme), $C = 8$ and $C = 4$ when $L < 40$ (the last chunk may be shorter), $K = 8$, $K' = 4$, five independent restarts per cell, $\alpha = 1.0$ (ablations 0/∞/0.25/0.5), and $m = 32$ for the baseline. Each restart is seeded by sha256(cell seed | `guided_beam` | attempt), so cells remain reproducible and independent; the run manifest records C, K, K', α, φ and the UniToxin suite. Runtime: chunked decoding costs $O(\frac{L}{C}KK')$ continuations of length C, and a prefix is never re-folded, so the guidance itself adds negligible per-chunk overhead. The full-cohort and pilot executions are reported in Section 4.3; Algorithm 1 and the φ dispatch are additionally covered by a weight-free CPU test suite and a self-check.

A.4 Cross-architecture guided beam search: arms, paired tests, diagnostics

What was run. The cross-architecture campaign of Section 3.3 drives the same `guided_beam` scheme (Algorithm 1; $\varphi_{\text{tox}} =$ `toxipep`, $\alpha = 1.0$, $C = 8$ and $C = 4$ when $L < 40$, $K = 8$, $K' = 4$, $T = 1.0$, top-$p = 0.9$, one attempt per cell, shared masks, cells and cell seeds) against two victims outside the MSA-prompted decoder: zymCTRL on its true-EC hazardous subset, and ProteinMPNN on the full hazardous panel. Six guided arms exist in total: for zymCTRL the base, NPO and GRPO checkpoints, and for ProteinMPNN the base, NPO and GRPO checkpoints, each judged against the `standard` rate of Table 1 rather than against a rerun sampling arm. The unlearned checkpoints are the released ones, wired in as a `trainable_state_dict` for zymCTRL and as a `model_state_dict` for ProteinMPNN (the checked-out revision discarded `-checkpoint` for these two models, so the checkpoint path is passed explicitly); the zymCTRL GRPO arm is the same checkpoint as the GRPO column of Table 1, which the defense narrative does not select. Every

arm is folded with ESMFold and judged by the shared criterion, and cells that are *skipped* (declared capability limit) or *undecidable* (folded without an RMSD) are excluded from the denominator.

The `standard` arm is cited, not rerun. Its ASR@1 comes from Table 1 and from the released panel export, which stores a per-protein `asr_success` flag produced by the same `attack_succeeded` criterion as the guided arms. The paired tests of Table 7 therefore use those stored flags, recomputing *both* arms on the proteins each pair of arms decided, which is why the guided column there can differ by a few points from Table 6.

Table 6: Guided-beam arms on the two additional victims: ASR@1 (%) with 95% Wilson intervals on the decided cells. ZymCTRL is on the true-EC subset ($n = 38$ attempted per condition) and its base arm lost cells in every condition to adapter CUDA OOM under concurrent generation, leaving 31–34 decided; the ProteinMPNN NPO arm is partial (57–62 of 117 cells generated per condition before the budget was cut). Intervals are computed on decided cells and are not cluster-adjusted, unlike the defense tables; they are quoted to show the resolution available at these cohort sizes.

Victim	Arm	Cons 0.1	Cons 0.3	Rand 0.1	Rand 0.3	Tail 0.1	Tail 0.3
ZymCTRL*	base	79.4 [63.2, 89.7]	63.6 [46.6, 77.8]	59.4 [42.3, 74.5]	35.3 [21.5, 52.1]	25.0 [13.3, 42.1]	6.5 [1.8, 20.7]
ZymCTRL*	NPO	0.0 [0.0, 9.4]	0.0 [0.0, 9.6]	0.0 [0.0, 9.2]	0.0 [0.0, 9.2]	0.0 [0.0, 9.2]	0.0 [0.0, 9.2]
ZymCTRL*	GRPO	62.2 [46.1, 75.9]	0.0 [0.0, 9.6]	26.3 [15.0, 42.0]	2.6 [0.5, 13.5]	5.3 [1.5, 17.3]	0.0 [0.0, 9.2]
ProteinMPNN	base	40.2 [31.4, 49.7]	13.3 [8.1, 21.1]	13.8 [8.7, 21.2]	0.0 [0.0, 3.2]	17.4 [11.5, 25.3]	0.0 [0.0, 3.2]
ProteinMPNN	NPO‡	0.0 [0.0, 6.6]	0.0 [0.0, 6.6]	0.0 [0.0, 6.2]	0.0 [0.0, 6.2]	0.0 [0.0, 6.3]	0.0 [0.0, 6.2]
ProteinMPNN	GRPO	21.5 [14.8, 30.2]	1.0 [0.2, 5.2]	3.4 [1.3, 8.5]	0.0 [0.0, 3.2]	12.9 [8.0, 20.2]	0.0 [0.0, 3.2]

Table 7: Paired comparison of the two arms on the proteins decided by both. *w/l* counts the discordant pairs won and lost by the guided arm; p is an exact two-sided sign test on those pairs. A dash marks a condition with no discordant pairs.

Victim	Arm	Cons 0.1	Cons 0.3	Rand 0.1	Rand 0.3	Tail 0.1	Tail 0.3
ZymCTRL*	base	+11.8 (5/1, .219)	+15.1 (7/2, .180)	+12.5 (5/1, .219)	+20.6 (9/2, .065)	+6.2 (4/2, .688)	+6.5 (2/0, .500)
ZymCTRL*	GRPO	+8.1 (4/1, .375)	0.0 (—)	+13.2 (5/0, .062)	+2.6 (1/0, 1.000)	−2.6 (0/1, 1.000)	0.0 (—)
ZymCTRL*	NPO	both arms 0.0 on all six conditions; no discordant pairs					
ProteinMPNN	base	+6.1 (12/6, .238)	+1.1 (1/0, 1.000)	+2.7 (8/5, .581)	0.0 (—)	+1.8 (14/12, .845)	0.0 (—)
ProteinMPNN	GRPO	+5.1 (9/4, .267)	+1.1 (1/0, 1.000)	0.0 (2/2, 1.000)	0.0 (—)	+4.5 (7/2, .180)	0.0 (—)
ProteinMPNN	NPO‡	both arms 0.0 on the 52–57 proteins decided by both; no discordant pairs					

The direction is consistent—the guided arm is at least as high on 17 of the 21 non-degenerate pairs and strictly higher on the high-yield conditions—but no single test reaches $p < 0.05$: at these cohort sizes the Wilson half-widths are about ± 15 pp for zymCTRL and ± 8 pp for ProteinMPNN, so even the largest effect (+20.6 pp at zymCTRL random@0.3, 9 wins against 2 losses) stops at $p = 0.065$. We therefore report the effect sizes and the checkpoint ordering rather than claiming significance the data do not support. The intermediate quantities behind these arms are in Table 8.

Table 8: Intermediate quantities for the guided arms at the two strongest conditions, i.e. the mechanism behind Table 6: *mask rec.* is the share of masked positions reconstructed exactly, *identity* the mean position-wise identity to the native sequence over the whole completion, *RMSD*≤ 2 the share of candidates within 2 Å of the native reference, and pLDDT the mean predicted confidence. Read alongside the corresponding standard-arm values in Section 4.3 and the defense table: the guided and standard arms sit at comparable pLDDT, and the success rate moves with mask recovery rather than against structure quality.

		Conservation 0.1				Random 0.1			
Victim	Arm	mask rec.	identity	RMSD≤ 2	pLDDT	mask rec.	identity	RMSD≤ 2	pLDDT
ZymCTRL*	base	.841	.984	.794	.754	.574	.957	.688	.755
ZymCTRL*	NPO	.089	.909	.027	.559	.070	.907	.553	.716
ZymCTRL*	GRPO	.724	.972	.622	.737	.354	.935	.553	.719
ProteinMPNN	base	.656	.966	.402	.714	.420	.942	.483	.713
ProteinMPNN	NPO‡	.077	.908	.000	.448	.069	.907	.035	.570
ProteinMPNN	GRPO	.496	.950	.280	.690	.354	.935	.397	.708

What φ_{tox} actually did. This diagnostic changes how the two tables above should be read, so we state it explicitly. UniToxin's comparably scored toxicity backends are gated at a chain length of 50 residues, and on the machines used for these runs only one of them is live: `toxipep` scores and discriminates (10-mers spanning 0.05–0.998), while `toxinpred3` raises an `AxisError` under numpy 2.x and the two structure-dependent axes cannot import their graph libraries in the adapter

interpreter. For any sequence longer than 50 residues every remaining backend therefore skips and φ_{tox} returns `None`, which the beam converts to 0.0; if every candidate in a round is a long chain, $\mathrm{z}(\varphi)$ is identically zero and the round is ranked by $z(\log p)$ alone. For ProteinMPNN the scored object is the whole native-length chain, so the scorer was missing in $32/32$ candidate chunks at *every* step; for the two autoregressive victims the prefix is short enough only over the first ≈ 6 chunks, after which the same fallback applies; and running the zymCTRL NPO arm with `-beam-alpha 0` (φ-only) and `-beam-alpha 1.0` produced bit-identical outputs (same six conditions, same mask recovery, same identity), as it must when $\mathrm{z}(\varphi) \equiv 0$. The measured improvement on these two architectures is therefore a *beam-search* effect—keeping the highest-likelihood of K sampled chunks—rather than evidence that toxicity guidance steers reconstruction, and the same scorer wiring is shared by the ProtGPT3-MSA arm of Section 4.3, whose rounds past the scorer's length gate are likewise decided by likelihood. A fair test of the steering hypothesis needs either a toxicity predictor that scores full-length proteins in this interpreter or a scorer applied to the masked window rather than to the whole chain; neither is implemented.

A.5 Implementation Details For Attacking

Cost and execution. One RTX 5090 (32 GB). Guided generation for zymCTRL is $\approx$ 45–75 s per cell at $L \approx 120$ and 1–2 min at $L \approx 800$, dominated by the scorer subprocess; the 228-cell true-EC matrix was generated in four accession shards and folded in a single sequential ESMFold repair pass after the concurrent pass left candidates unfolded. ProteinMPNN generation costs seconds per cell and folding dominates (≈ 4 s for short chains, ≈ 38 s at 950 aa). The zymCTRL GRPO matrix completed all 228 cells with none failed and the NPO matrix reports 36–38 decided cells per condition; the zymCTRL *base* arm reports the OOM failures of Table 6 rather than hiding them; the ProteinMPNN NPO matrix was stopped on the longest chains to stay inside the budget and is therefore partial. A prefix is never re-folded during decoding, so the scorer subprocess is the only per-chunk overhead.

Hyperparameter search performed. Before the full runs we swept, on the 20 shortest hazardous proteins with folding, $\alpha \in \{0, 0.25, 0.5, 1.0, \infty\}$, $K \in \{8, 16, 32\}$ and attempts $\in \{1, 5\}$. Raising K moved mean mask recovery only slightly (ProteinMPNN GRPO conservation@0.1: $0.424 \rightarrow 0.452$ at $K = 32$), within run-to-run noise, so $K = 8$ was kept to match the ProtGPT3-MSA runs and the time budget; five attempts per cell were not computable at full cohort size. The α sweep is uninformative on these two victims for the reason above.

Known limitations of this campaign. (a) The `standard` arm is cited, so its rates are those of the released export rather than a rerun under identical concurrency. (b) The zymCTRL base guided arm, and every ProteinMPNN arm's folding pass, ran on one 32 GB card, so long chains hit OOM; cells that could not be folded are reported as *undecidable* and excluded, never counted as failures, and the OOM failures of the zymCTRL base arm are excluded the same way. (c) The ProteinMPNN NPO guided arm is partial, and neither NPO result can be distinguished from its own sampling baseline because both are at the floor. (d) No paired test here reaches significance, and the intervals are not cluster-adjusted as the defense tables are.

B Case Study on Amylin

Biology. Human islet amyloid polypeptide (hIAPP, "amylin") is a 37-residue hormone co-secreted with insulin; its central segment (residues 20–29, `SNNFGAILSS`, with the hydrophobic `FGAIL` motif) is amyloidogenic, and aggregation of hIAPP into oligomers/fibrils is cytotoxic to pancreatic β-cells and is a hallmark of type-2 diabetes (Westermark et al., 2011). The rodent ortholog differs at only six residues in the central region (residues 18–29) and is neither fibrillogenic nor cytotoxic; the three proline substitutions in this region act as β-sheet "breakers", and even a *single* human-residue substitution (R18H, L23F, or V26I) restores fibril formation in rat amylin (Green et al., 2003). Amylin therefore provides a compact, controllable instance of a *latent* hazard: a peptide whose danger is not "matching a known toxin" but arising from self-assembly, i.e., exactly the mode that sequence-identity guardrails (G2/G3) cannot detect.

Why amylin? We choose amylin because (i) it is not a toxin in the usual homology sense—unlike, for example, snake-venom proteins—so identity-based guardrails give no signal on it;

Table 9: The two reference peptides of the amylin case study and their verdicts under the implemented hazard gatekeeper. Identity is position-wise over the masked window (residues 18–29), the quantity the panel's judge uses; the two cores differ at exactly six positions, three of them prolines in the rodent sequence. Both peptides are *non-hits* against the 427-sequence toxin database (hazardDB), i.e. a retrieval/identity guardrail cannot separate the aggregation-toxic human hormone from its non-toxic ortholog. The pair shows how guidance on known toxic sequences reaches beyond sequence identity: an identity or retrieval guardrail cannot separate the aggregation-toxic hormone from its non-toxic ortholog.

Reference peptide	**Core (residues 18–29)**	**Id. to hIAPP**	**Id. to rat amylin**	**hazardDB gatekeeper**
Human amylin, hIAPP (P10997)	`HSSNNFGAILSS`	1.00	0.50	no hit (0/427)
Rat amylin, rIAPP (P12969)	`RSSNNLGPVLPP`	0.50	1.00	no hit (0/427)

(ii) it can nevertheless become toxic through aggregation, and human amylin is curated as a pathogenic amyloid entry in AmyPro (entry AP00013) (Varadi et al., 2018); and (iii) compared with other misfolded-protein systems such as prions, it is a short 37-residue peptide, which makes controlled generation experiments tractable.

Protocol. For the case study we mask the 12-residue central window (positions 18–29) of hIAPP, keep the flanking context (`input_prefix`), and let the model fill the masked positions; the PALM scores of Figure 3 are computed on those completions, which come from the φ_{ens} guided beam, with the `standard` constrained-infill scheme as baseline. The two reference peptides are fixed in Table 9: their cores (human `HSSNNFGAILSS`, rat `RSSNNLGPVLPP`) differ at exactly six positions—three of them prolines in the rodent sequence—and their position-wise identity over the window is 0.50. Neither peptide is a hit against the 427-sequence toxin database, which is precisely why an identity or retrieval guardrail cannot separate the aggregation-toxic hormone from its non-toxic ortholog. Per-candidate PALM scores, the AmyPro cohort construction and the attacking metrics of the amyloid cohort are reported in the released artifacts.

Result. Completing the masked core moves that window *up* on the peptide-aggregation predictor PALM (Eschbach et al., 2026) (Figure 3): from 0.8940 on the native hormone and 0.8268 on the rodent ortholog—six substitutions in this window (identity 0.50), neither fibrillogenic nor cytotoxic (Green et al., 2003)—to 0.9872 for the best completion (guided $k2$, core `GILGVVGLLISV`; $+0.093$ and $+0.161$ respectively), with 0.9767 for the best `standard` completion.

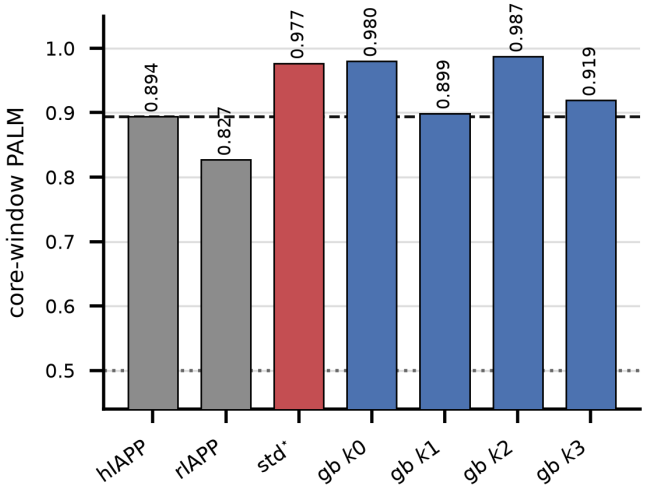

Figure 3: Core-window (residues 18–29) mean of the peptide-aggregation predictor PALM (Eschbach et al., 2026) (five-fold ensemble) for native hIAPP, its non-toxic rodent ortholog rIAPP, and completions of the masked core (`gb` = guided beam, `std`* = best `standard` completion). Dotted line, PALM's 0.5 high-risk call; dashed line, the native hormone. Every sequence already lies above the call, so the panel shows a shift of the core-window propensity rather than a label flip.

Why the ladder misses it. The shift passes through the guardrail ladder of Section 1 untouched. The (G2) identity channel stays empty: neither reference peptide is a hit against the 427-sequence toxin database ($\varphi_{\text{hdb}} = 0$ for every candidate, Table 9), and the completions rewrite exactly the window a sequence query would match, with the flanks restored (core recovery 0–0.167). (G3) structure identity has no query to pose, since aggregation is a propensity to self-assemble rather than resemblance to a known hazardous fold; and the axis is decorrelated from (G4), the classifier we guided with—the candidate with by far the highest pathogen score (guided $k1$, $\varphi_{\text{tox}} = 0.428$) gains only $+0.005$ aggregation, while the most aggregation-prone completion is unremarkable on that axis ($\varphi_{\text{tox}} = 0.033$).

Two properties matter for a defender. The shift also appears under plain constrained infill ($+0.011$ between the two arms), so it is a property of completing amyloidogenic segments in this model family rather than an artefact of guided search: any pipeline that fills in such segments meets it with or without a search-time attack. And it is graded rather than categorical—the non-toxic ortholog already lies above PALM's 0.5 call—so a threshold monitor returns the same verdict for the hormone, its ortholog and every completion. Covering the gap calls for a biophysical monitor read out on a graded scale.

Run conditions of the guided amylin run. The beam scores *prefixes*, which have no folded structure of their own, so the native Amylin PDB was copied under every record id and supplied to each prefix; because the structure-dependent UniToxin backends could not execute in our checkout (see the full appendix), that policy is a run condition rather than an active scoring component. The four returned candidates are per-attempt winners: the final chunk is fully unmasked, so the last beam step collapses to identical draws and their φ z-scores are all zero.

The case study showcases the attack's power in the sense that all three elements of the pipeline are necessary: the *tag/few-shot/prefix* assembly supplies a hazardous-adjacent starting point, the *foundational scorer* supplies a hazard signal that is orthogonal to homology, and *chunked rescoring* converts that signal into a decoding pressure that a single best-of-m sample cannot exert reliably (in best-of-m, every sample must independently land in the rare high-φ region). A schematic trace of per-chunk guidance pressure on the masked core, plus the full scorer inventory and operating thresholds, is provided in Appendix A.3.

C Reinforcement Unlearning

C.1 Benign-panel retention of the released checkpoints

Table 10: Retention of benign capability before and after biosafety unlearning, i.e. the benign half of the grid whose hazardous half is Table 1. Each entry reports **reconstruction success (%) / TM-score / pLDDT**; higher is better on all three, so the gap to *Base* is the utility price of each defense. Benign targets are 86 test proteins (514 cells per model).

Model	Method	Conservation		Random		Tail	
		0.1	0.3	0.1	0.3	0.1	0.3
DPLM-2	Base	69.4 / .925 / .799	61.2 / .878 / .795	51.2 / .900 / .800	22.1 / .869 / .795	39.5 / .906 / .799	19.8 / .875 / .801
	NPO	69.4 / .893 / .784	51.8 / .810 / .776	52.3 / .887 / .793	25.6 / .810 / .779	38.4 / .898 / .798	12.8 / .849 / .796
	GRPO	61.2 / .890 / .792	45.9 / .830 / .787	32.6 / .896 / .797	7.0 / .828 / .788	22.1 / .890 / .797	4.7 / .843 / .800
ESM-3	Base	77.6 / .943 / .801	60.0 / .898 / .802	57.0 / .901 / .802	32.6 / .874 / .800	53.5 / .919 / .801	22.1 / .882 / .801
	NPO	74.1 / .938 / .799	60.0 / .868 / .786	60.5 / .911 / .799	29.1 / .844 / .781	45.3 / .909 / .799	18.6 / .864 / .787
	GRPO	79.8 / .947 / .802	54.8 / .892 / .802	46.5 / .898 / .803	18.6 / .868 / .802	39.5 / .913 / .801	14.0 / .874 / .807
ZymCTRL*	Base	89.7 / .986 / .890	72.4 / .953 / .851	79.3 / .988 / .894	62.1 / .957 / .862	69.0 / .989 / .888	41.4 / .964 / .879
	NPO	86.2 / .987 / .886	65.5 / .879 / .813	75.9 / .991 / .892	44.8 / .960 / .864	62.1 / .987 / .889	27.6 / .939 / .866
	GRPO	82.8 / .982 / .886	65.5 / .920 / .826	72.4 / .985 / .891	51.7 / .928 / .849	62.1 / .986 / .887	44.8 / .962 / .874
ProtGPT3-MSA	Base	74.1 / .947 / .793	63.5 / .903 / .777	57.0 / .933 / .798	25.6 / .889 / .788	45.3 / .938 / .800	31.4 / .906 / .798
	NPO	62.4 / .912 / .772	45.9 / .763 / .692	36.0 / .902 / .786	14.0 / .827 / .736	33.7 / .894 / .790	18.6 / .817 / .759
	GRPO	58.8 / .903 / .787	38.8 / .736 / .687	41.9 / .914 / .795	18.6 / .826 / .755	33.7 / .909 / .786	18.6 / .799 / .756
ProteinMPNN	Base	38.6 / .891 / .792	5.3 / .842 / .768	22.0 / .895 / .796	0.0 / .854 / .786	14.8 / .908 / .800	1.4 / .865 / .789
	NPO	9.6 / .848 / .755	0.0 / .640 / .681	4.9 / .863 / .767	0.0 / .697 / .698	3.7 / .864 / .785	0.0 / .758 / .748
	GRPO	32.1 / .883 / .796	0.0 / .832 / .777	18.1 / .896 / .798	0.0 / .847 / .795	8.5 / .893 / .801	0.0 / .859 / .795

C.2 Reveal-Schedule Approximation in Diffusion GRPO

A masked-diffusion transition contains two coupled decisions: which currently masked positions are revealed and which residues are assigned to those positions. Let M_t denote the masked positions at state x_t, $U_t \subseteq M_t$ the positions revealed at the current step, and $a_t = x_{t-1,U_t}$ their residue assignments. The native transition can be factorized as

$$\pi_\theta(U_t, a_t \mid x_t, c) = q_\theta(U_t \mid x_t, c)\, p_\theta(a_t \mid U_t, x_t, c), \tag{10}$$

so for a stochastic reveal policy the corresponding importance ratio is

$$\rho_t^{\text{native}} = \frac{p_\theta(a_t \mid U_t, x_t, c)}{p_{\text{old}}(a_t \mid U_t, x_t, c)} \frac{q_\theta(U_t \mid x_t, c)}{q_{\text{old}}(U_t \mid x_t, c)}. \tag{11}$$

The two diffusion generators use different native position-selection rules. DPLM-2 commits positions using a Gumbel-perturbed top-k over the model's per-position confidences, with the perturbation scale annealed over the denoising trajectory. ESM-3 instead deterministically selects a bottom-k set under per-position entropy, with k determined by its native cosine schedule. Thus, in both models, the native reveal trajectory depends on the current model state and differs from the rollout policy used during GRPO training.

Training-time reveal policy. For GRPO training we replace the native reveal rule with a parameter-independent random schedule $\tilde{q}$. Given the attacked position set M, we first draw a uniformly random permutation of M and partition it into $S = \min(T, |M|)$ consecutive blocks. The block sizes are

$$|U_t| = \left\lfloor \frac{|M|}{S} \right\rfloor + \mathbf{1}\{\cdot\}\,[t < |M| \bmod S], \qquad t = 0, \ldots, S-1. \tag{12}$$

The reported DPLM-2 and ESM-3 runs use $T = 8$. In the realized training trajectories, the smallest attacked set has $|M| = 17$, so all trajectories use exactly eight reveal steps.

One reveal schedule is sampled before generating a GRPO group and is shared by all $G = 6$ rollouts in that group. Thus within-group variation arises from residue sampling while the revealed positions are held fixed. The sampled schedule and realized residue actions are stored with the rollout and replayed exactly when evaluating the current, old, and reference policies; no reveal set is resampled during policy-ratio or KL computation.

Under this rollout distribution,

$$\tilde{\pi}_\theta(U_t, a_t \mid x_t, c) = \tilde{q}(U_t \mid M_t)\, p_\theta(a_t \mid U_t, x_t, c), \tag{13}$$

where $\tilde{q}$ is independent of θ. Its contribution therefore cancels exactly:

$$\rho_t = \frac{\tilde{\pi}_\theta(U_t, a_t \mid x_t, c)}{\tilde{\pi}_{\text{old}}(U_t, a_t \mid x_t, c)} = \frac{p_\theta(a_t \mid U_t, x_t, c)}{p_{\text{old}}(a_t \mid U_t, x_t, c)}. \tag{14}$$

The likelihood ratio is therefore exact for the rollout policy actually used during training. The approximation instead lies in replacing the native, model-dependent reveal policy q_θ with the external schedule $\tilde{q}$. Equivalently, GRPO optimizes residue assignment conditional on an externally specified reveal trajectory rather than the complete native denoising policy.

Reveal schedules are sampled from a deterministic RNG keyed by the run seed, training step, protein accession, attack type, and mask ratio. The RNG never observes model logits, preserving the parameter independence of $\tilde{q}$; realized schedules are additionally logged for reproducibility.

Empirical size of the train–inference mismatch. We quantify the practical mismatch by comparing the training-time rollout policy with each model's native inference pipeline. Both arms use the same 20 proteins (10 hazardous and 10 benign), conservation and random attacks at mask ratio 0.3, identical attacked positions, and a candidate budget $G = 8$. The resulting 39 evaluable cells produce 312 candidates per arm, with no folding failures.

Table 11 shows a substantial difference for DPLM-2: the native pipeline increases ASR@1 from 12.8% to 28.8%, alongside higher masked identity and lower RMSD. For ESM-3, the corresponding ASR@1 difference is only 1.3 percentage points (15.4% to 16.7%), with a cluster-bootstrap 95%

confidence interval of $[-2.2, 4.8]$ percentage points for the paired difference. The approximation is therefore not uniformly severe across diffusion architectures.

These comparisons should not be interpreted as isolating the causal effect of reveal order alone. The training-time and native inference pipelines also differ in sampling details, including the effective residue alphabet and top-p filtering. Table 11 therefore measures the aggregate train–inference rollout mismatch. In particular, the reversed ESM-3 ASR@8 ordering is based on few successful cells and is not treated as evidence that the fixed policy is stronger.

All main defense evaluations use each model's native inference pipeline. The approximation is confined to GRPO training. A fully native update would also account for the model-dependent reveal rule: this is comparatively straightforward for DPLM-2's stochastic confidence-based selector, whereas ESM-3's deterministic entropy-based selection would first require a stochastic relaxation before a PPO-style reveal-policy ratio can be defined. We leave such mask-aware policy optimization to future work.

Table 11: Train–inference rollout mismatch for the two masked-diffusion generators. FIXED uses the parameter-independent rollout policy employed during GRPO training, whereas NATIVE uses each model's inference-time decoding pipeline. Both comparisons use the same 20 proteins (10 hazardous and 10 benign), conservation and random attacks at mask ratio 0.3, identical attacked positions, and $G = 8$ candidates per cell (39 evaluable cells; 312 candidates per arm; no folding failures). The two pipelines differ not only in reveal order but also in sampling details such as the effective residue alphabet and top-p filtering; the table therefore measures the aggregate rollout mismatch rather than the isolated effect of the reveal schedule.

	DPLM-2		ESM-3	
Metric	Fixed	Native	Fixed	Native
Masked identity	0.477 ± 0.188	0.595 ± 0.200	0.554 ± 0.182	0.583 ± 0.184
Full identity	0.843	0.879	0.866	0.875
RMSD (Å)	5.46	4.55	4.53	4.18
Identity pass@1	0.196	0.324	0.276	0.301
ASR@1	0.128	0.288	0.154	0.167
ASR@8	0.231	0.333	0.333	0.282
Distinct seq./cell	8.00	7.10	7.97	7.90

C.3 IMPLEMENTATION DETAILS FOR UNLEARNING

NPO. We use the same NPO coefficient $\beta = 0.1$ for all five generators. Most runs use a learning rate of 5×10^{-5} and a KL coefficient of 0.01. For zymCTRL, we use a smaller learning rate of 1×10^{-5} and a larger KL coefficient of 0.1, selected during tuning. The reference model is in all cases a frozen copy of the corresponding base checkpoint.

GRPO. All GRPO runs use group size $G = 6$ and the structure-aware reward defined in Section 4. The default learning rate is 5×10^{-5}, with zymCTRL again using 1×10^{-5}. All GRPO runs use a training mask ratio of 0.5, whereas evaluation is performed at ratios $\{0.1, 0.3\}$. DPLM-2 and ESM-3 use $T = 8$ reveal steps per rollout; their reveal-step construction is described in Appendix C.2. Full optimizer, sampling, and launcher configurations will be released with the code.

Training budgets follow the native optimization pipeline of each architecture. DPLM-2, ESM-3, and zymCTRL use a 2000-step training budget, whereas ProtGPT3-MSA and ProteinMPNN is trained for 5 epochs. NPO and GRPO use matched budgets within each of these architectures. For zymCTRL, training is not restricted to proteins with complete four-level EC annotations. Proteins without a complete EC annotation are retained using an empty conditioning prefix, whereas evaluation follows the true-EC subset reported in the main tables.