SAFEGENIE: Erasing Dangerous Concepts from Biological Diffusion Models

Anonymous Author(s)
Affiliation
Address
email

Abstract

Generative diffusion models have rapidly advanced protein design, but their flexibility introduces biosafety risks: the same models that scaffold therapeutic enzymes can also produce prions, toxins, or other harmful proteins. Post-hoc defenses like filters and classifiers are brittle and vulnerable to jailbreak-style prompting. We introduce SAFEGENIE[1], a weight-level erasure framework that reshapes the model's probability distribution to proactively suppress unsafe concepts, making the resulting generators resilient to inference-time attacks. Through targeted experiments, we show that SafeGenie can reduce the likelihood of generating structural motifs such as α-helices, eliminate prion-like aggregation signals, and lower toxic peptide predictions, all while preserving designability and diversity. We further construct a unified SafeGenie model by erasing 1,450 PDB-labeled toxins, demonstrating that large-scale distributional erasure yields a generator that reliably avoids unsafe sequences without degrading overall protein quality. Our results establish weight-space probability editing as a principled, robust, and practical tool for biosafety in generative biology.

1 Introduction

1.1 Motivation

Generative protein diffusion models have enabled precise control over protein backbones and scaffolding of functional motifs, drastically reducing the time for de-novo protein design. In general, models such as Genie [Lin and AlQuraishi, 2023], RFDiffusion [Watson, 2023], and Chroma [Ingraham et al., 2023] have been used positively to accelerate enzyme engineering, stabilize therapeutic proteins, and generate novel binders with high affinity [Zambaldi et al., 2024].

However, this flexibility introduces new safety concerns. Generative targeted models may produce harmful proteins either deliberately, such as the case of engineering neurotoxins for high-affinity binding, or unintentionally by introducing proteins that can misfold into prion-like structures or disrupt other bodily functions. Recent work has shown that these same models can also pose significant biosafety risks, such as producing sequences with strong similarity to known toxins or generating harmful dual-use proteins like membrane disrupters. Although there have been many recent calls for policy regulation of these biological generative models Pannu et al. [2025], Hunter et al. [2024], the rate of development often exceeds the rate of regulation, which necessitates the need for model-based safety measures.

[1]All code and data will be made available after the double-blind review process is concluded

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

1.2 Contributions

Our work makes the following contributions:

1. We introduce SAFEGENIE, the first framework for erasing concepts from protein diffusion models. Unlike post-hoc filtering, SAFEGENIE erases concepts directly at the weight-level through a distribution erasure objective, addressing safety concerns before generation happens.
2. Through case studies, we demonstrate successful suppression of both global structural features (e.g. α-helices) and pathogenic motif-level features (e.g. prion-like domains).
3. We present a Unified SAFEGENIE Model trained on toxic proteins from a PDB and evaluate it with protein design benchmarks (designability, diversity, F1) as well as toxicity prediction pipelines, showing that erasure reduces unsafe generations while preserving statistically-identical high-quality protein generation capabilities.

1.3 Related Works

Generative Protein Models: Backbone-first generative models have rapidly advanced protein design by directly sampling 3D structures prior to sequence realization. RFdiffusion introduced a denoising diffusion framework that conditions on functional motifs and scaffolds novel backbones with high success rates [Watson, 2023]. Genie extends this idea with SE(3)-equivariant diffusion over oriented residue frames, producing diverse and designable structures [Lin and AlQuraishi, 2023]. Chroma leverages score-based generative modeling with symmetry-aware networks to sample backbones while enabling fine-grained conditional control [Ingraham et al., 2023]. DiffDock adapts diffusion methods for ligand-conditioned backbone generation, demonstrating flexibility in drug discovery contexts [Ketata et al., 2023]. Other approaches such as FrameDiff [Yim et al., 2023], experiment-guided diffusion hybrids [Liu et al., 2024], and flow-based geometry generators for protein ensembles [Jing et al., 2024] highlight a growing ecosystem where geometric priors and equivariance play central roles. Collectively, these backbone-first methods represent a paradigm shift from sequence-only generative models by enabling explicit geometric control, active site scaffolding, and the design of folds absent from the natural repertoire.

Safety In Generative Biology Models: Generative biology research has begun to incorporate biosafety measures, yet these remain nascent and incomplete. For example, SafeProtein introduced a systematic red-teaming approach along with a benchmark (SafeProtein-Bench), which demonstrates that protein foundation models such as ESM3 and DPLM2 can be 'jailbroken' with great success using masked prompt strategies and beam search, thus revealing that current models remain vulnerable despite dataset filtering [Fan et al., 2025]. Similarly, FoldMark proposes embedding watermarks into outputs of protein generative models (including diffusion-based models such as RFDiffusion and FrameDiff) in order to trace misuse. However, watermarking merely ensures traceability, not prevention of harmful sequence generation or intentional obfuscation [Zhang et al., 2024]. Finally, recent evaluations on inference time filters find that they often fail to detect known viral-host interactions, much less novel threats, highlighting that post-hoc filters are unreliable for biosafety [Feldman and Feldman, 2025]. Together, while these methods mark early progress, they do not sufficiently prevent model misuse. Red-teaming exposes weaknesses rather than fixes them, watermarking does not stop hazardous output, DNA jailbreak frameworks reveal scalability of risks, and current filters frequently miss even known dangerous proteins.

Removing or Mitigating Harmful Capabilities in Models: Beyond detection and filtering, a growing body of research in machine learning explores how to directly suppress, remove, or steer away from dangerous capabilities within generative models. Unlike post-hoc defenses, these approaches aim to proactively alter what a model can represent or output, thereby reducing the risk of misuse even under adversarial prompting. One prominent line of work focuses on the direct removal of information from model weights through methods such as lightweight erasers [Huang et al., 2024], unified closed-form edits [Gandikota et al., 2024], and precise single-concept deletion [Gandikota et al., 2023], which modify internal parameters to eliminate targeted concepts in a way that prevents easy reintroduction.

Such erasure methods are particularly appealing because they are inherently robust to jailbreak-style attacks, do not rely on brittle post-hoc filtering, and can in principle be shipped safely without exposing

dangerous capabilities. By removing harmful knowledge at the parameter level, these approaches offer protection not only against intentional misuse (e.g., adversaries seeking to elicit toxic proteins) but also against accidental harms, where a model might generate hazardous sequences in the course of legitimate use. While these methods have primarily been deployed in vision and text domains, they highlight a promising paradigm for biosafety in protein diffusion models, where eliminating dangerous motifs or folds at their representational source could provide stronger guarantees than detection-based defenses alone.

2 Methods

2.1 Training Objective

We apply erasure to Genie2 [Lin et al., 2024], a diffusion process over the Cartesian coordinates of the N central C_α atoms of a given protein. A sample protein $\mathbf{x_0}$ is selected from the protein structure distribution, and then isotropic Gaussian noise is added following a standard cosine variance schedule: $\beta = [\beta_\mathbf{1}, \ldots, \beta_\mathbf{t}]$. By the reparameterization trick [Ho et al., 2020], we can represent the forward process at timestep t as:

$$q(\mathbf{x}_t \mid \mathbf{x}_{t-1}) = \mathcal{N}\big(\sqrt{\alpha_t}\,\mathbf{x}_{t-1},\ (1-\alpha_t)\,\mathbf{I}\big), \qquad q(\mathbf{x}_t \mid \mathbf{x}_0) = \mathcal{N}\big(\sqrt{\bar{\alpha}_t}\,\mathbf{x}_0,\ (1-\bar{\alpha}_t)\,\mathbf{I}\big) \tag{1}$$

And the backward process as, using $\alpha_t := 1 - \beta_t$ and $\bar{\alpha}_t := \prod_{s=1}^{t} \alpha_s$:

$$q_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t) = \mathcal{N}\left(\frac{1}{\sqrt{\alpha_t}}\Big(\mathbf{x}_t - \tfrac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\,\epsilon_\theta(\mathbf{x}_t, t)\Big),\ \sigma_t^2\mathbf{I}\right), \quad \sigma_t^2 = \tilde{\beta}_t := \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\,\beta_t \tag{2}$$

The backward process requires a noise prediction, $\epsilon_\theta(\mathbf{x}, t)$, which is generated through an SE(3) equivariant denoiser network. This denoiser is comprised of two linear networks (a single feature network with weights θ_{SFN} and a pair feature network with weights θ_{PFN}) and one transformer layer (a pair transformer network with weights θ_{PTN}), whose sets of weights we can define as $\theta = \{\theta_{SFN}, \theta_{PFN}, \theta_{PTN}\}$.

In general, we can view the denoiser as a high-dimensional manifold projector that guides noisy coordinates back towards the distribution of valid protein conformations Abuduweili et al. [2024]. Armed with the view that the denoiser controls the sampling distribution, we can imagine 2 different SE(3) equivariant denoisers, one that generates an undesirable probability distribution with weights θ^* and another that generates a desirable distribution with weights θ.

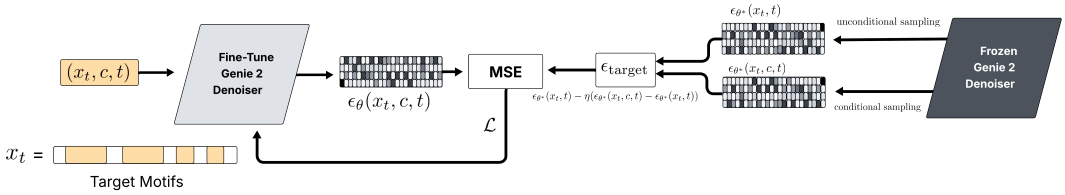


Figure 1: Fine Tuning Architecture

Ideally, we'd like to be able to assign a low probability of generation to undesired concepts in our distribution. To do this, we follow Gandikota et al. [2023], reducing the probability of generating a specific output x given by the likelihood of it being described by the concept c according to a power law $P_{\theta(x)} = \frac{P_{\theta^*}(x)}{P_{\theta^*}(c|x)^\eta}$.

After following Bayes Rule, taking gradients of the log probability, and applying Tweedie's formula to introduce time varying noise, we can relate the noise prediction of θ and θ^* such that they follow the power law:

$$\epsilon_\theta(x_t, t) = \epsilon_{\theta^*}(x_t, t) - \eta(\epsilon_{\theta^*}(x_t, c, t) - \epsilon_{\theta^*}(x_t, t)) \tag{3}$$

A full derivation of this equation can be found in Appendix A. We pose this relationship as our objective to optimize over, training θ to minimize the mean-square-error difference between its current predictions and the power-law steering prediction, as depicted in Figure 1. To enable scale-equivalence, we tag on an additional weighting term so that the current predictions do not have a larger magnitude than the η guided power law prediction:

$$\mathcal{L} = ||\epsilon_\theta(x_t, t) - \frac{\epsilon_{\theta^*}(x_t, t)}{\epsilon_{\text{target}} + 10^{-8}} \epsilon_{\text{target}}||_2 \tag{4}$$

$$\epsilon_{\text{target}} = (\epsilon_{\theta^*}(x_t, t) - \eta(\epsilon_{\theta^*}(x_t, c, t) - \epsilon_{\theta^*}(x_t, t))) \tag{5}$$

2.2 Training Details

We fine-tune Genie2 by wrapping the SE(3)-equivariant denoiser into a PyTorch Lightning module on a single A1000. Fine-tuning is performed against a frozen reference model θ^*, which provides the teacher predictions for the concept-adjusted noise targets. We optimize parameters θ using the Adam optimizer with a learning rate of 1×10^{-5} and weight decay of 1×10^{-4}. We optimize **all** weights θ, rather than a subset; Appendix B reports ablation studies on weight choice and describes how we determined which parameters to optimize. To stabilize training, we apply gradient clipping with a maximum ℓ_2-norm of 1.0 across all trainable parameters.

For efficiency, we accumulate gradients over a scaffolds before each optimizer update. Specifically, each training step samples a random scaffolds, computes the masked mean-squared-error loss on each, and backpropagates. Losses are normalized by a, and gradients are accumulated across these passes before a single optimizer step. In our experiments, we set $a = 4$. At every step, we (i) randomly select a motif scaffolding problem file, (ii) construct conditioned features (fixed residues corresponding to the motif) and unconditioned features (motif mask zeroed out), and (iii) add isotropic Gaussian noise at a randomly sampled diffusion timestep t. To ensure valid structural signals, motif residues are preserved in their original positions while only scaffold residues are perturbed. Unless otherwise specified, we fine-tune for 1000 optimization steps, each consisting of $a = 4$ accumulated scaffolds and $s = 4$-6 noisy samples per scaffold. Sequence lengths are randomly drawn between $L_{\min} = 150$ and $L_{\max} = 256$ residues. This stochastic batching encourages robustness across scaffold sizes while preserving fixed motifs.

3 Erasure Case Studies

To demonstrate the capability of the erasure algorithm (4), we erase a common structural motif from the distribution entirely (alpha helices) as well as targeting a specific unsafe motif (amyloids) and erasing it.

3.1 Alpha Helix Erasure

Training: We select 15 proteins of length 256 or less from the TMalphaDB database [Perea et al., 2015], extract the alpha-helices, and optimize over them using the objective function in the previous section. We fine-tune 4 values of η, $\eta \in \{0, 0.5, 1, 10\}$, and compare the results to the base Genie2 model. A detailed experimental procedure, hyperparameter information, and loss plots can be found in Appendix C

Evaluation: Figure 2 reports the fraction of generated proteins containing any alpha helix when samples 20 times. The baseline model reproduces helices in all sampled generations, consistent with its training distribution. Introducing even a modest erasure penalty ($\eta = 0.5$) reduces alpha helix prevalence to 90.4%, while strong erasure ($\eta = 10$) eliminates helices altogether. This demonstrates that the erasure signal is both effective and tunable: stronger penalties monotonically decrease the likelihood of generating the targeted concept.

Figure 3 analyzes helix length distributions. At $\eta = 0$, the model recapitulates naturalistic helix lengths centered around 20 residues. For $\eta = 0.5$, the distribution is compressed but not abolished, indicating partial suppression. At $\eta = 10$, the distribution collapses to near zero mass, confirming complete motif removal. This graded suppression provides evidence that erasure is not a brittle intervention, but rather admits fine-grained control over structural frequencies.

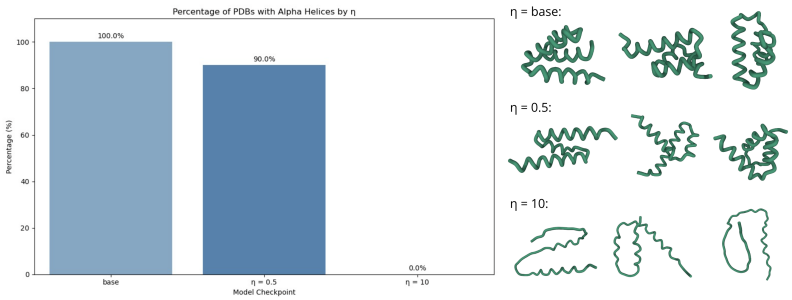


Figure 2: (Left) Percent of Generations With Alpha Helices, (Right) First 3 Proteins Generated

The erasure weight η directly governs the strength of the penalty applied to the targeted concept, and thus controls the balance between suppression and preservation of structural features. At low values ($\eta \leq 0.5$), the penalty is weak relative to the base generative prior, leading to partial erasure. This regime highlights that the model retains some inductive bias toward producing helices, but their prevalence and average size are measurably reduced. As η increases, the penalty term dominates the objective, and the model progressively reconfigures its generative distribution to avoid helices altogether. At $\eta = 10$, the near-complete elimination of helices suggests that the optimization landscape allows strong penalties to override even deeply embedded structural motifs, like alpha-helices. However, this comes at a cost: we observe broader distributional shifts in non-targeted structural features, suggesting that high η values can induce spurious correlations or degrade generalization.

This trade-off illustrates a general principle of concept erasure in high-dimensional models: small η values yield interpretable *attenuation* of the target concept without large off-target effects, while large η values produce stronger suppression but risk unintended distributional drift. Thus, η should be interpreted not as a binary "erase vs. preserve" switch, but as a continuous knob controlling the degree of structural editing and its side effects.

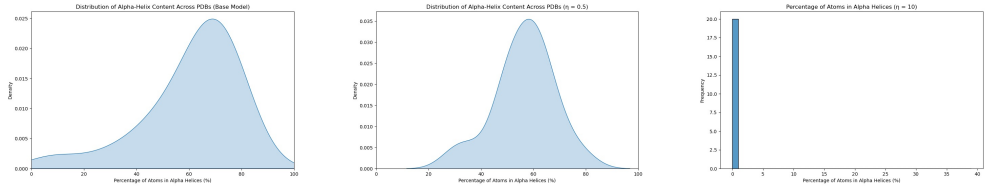


Figure 3: Comparison of Alpha Helix Size distribution for parameters 0, 0.5, and 10.

3.2 Prion Erasure

Prions are misfolded proteins that cause neurodegenerative diseases by inducing normal versions of the same protein to adopt their abnormal conformation [Colby and Prusiner, 2011]. Due to the dangerous nature of these proteins, we would like to condition our generative models to avoid creating prion-like outputs.

Training: To erase this concept, we select 4 prions from the Protein Databank and condition (4) on the entire sequence for each protein. We then train the model at $\eta = 5, 10$ for 450 steps; the full training details can be found in Appendix D.

Evaluation: We use the Modified Prion Aggregation Prediction Algorithm (mPAPA) to identify prion-like domains (PrLDs) in protein sequences based on amino acid composition and aggregation propensity [Cascarina and Ross, 2020, Toombs et al., 2010]. By design, the mPAPA metric returns a value of -1 if no intrinsically disordered segments are detected within a protein. Scores greater than -1 indicate the relative likelihood that a protein contains prion-like characteristics.

Using mPAPA, we find the protein's predicted window of amino acids contributing the most to its classification as a prion, and then remove that window of amino acids from the protein. We then

used this masked protein as a motif to conditionally generate outputs from the base Genie-2 model, a finetuned Genie model with $\eta = 5$, and a finetuned Genie model with $\eta = 10$. Figure 4 demonstrates model performance on testing with the first 59 sequences of protein Human prion protein variant M166V [Calzolai et al., 2000]. We hope to extend this framework to a generalized benchmark in future works.

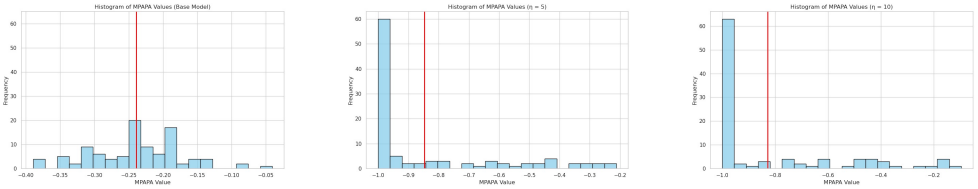


Figure 4: Comparison of mPAPA score distributions for parameters 0, 5, and 10.

Figure 4 highlights how increasing the regularization parameter η sharpens the prion-discrimination, collapsing variance. The base model ($\eta = 0$) yields a broad distribution centered near -0.25, capturing heterogeneous sequence-level variation, while $\eta = 5, \eta = 10$ progressively concentrate scores at -1 with sparse excursions, effectively driving most proteins into the "non-prionic" regime. This distributional collapse suggests that higher η values impose stringent penalties on folding disorder, suppressing borderline prion-like domains and biasing the model toward conservative predictions.

4 A Unified SafeGenie Model

Training: To develop a unified safe model, we erase the 1450 proteins labeled "TOXIN" in the PDB under length 256. We do this by setting the entire protein as the motif to erase, and then updating the model parameters θ^* using (4) for 350 steps. We create 2 variants, $\eta = \{5, 10\}$, denoted by SafeGenie-η. A complete discussion of training data, model parameters, and loss curves can be found in Appendix D.

Evaluation: To evaluate the toxicity of the model, we first generate protein samples from the base model, SafeGenie-5, and SafeGenie-10. For each sample, we generate 5 likely sequences given the backbone using proteinMPNN [Dauparas et al., 2022], and then use ToxinPred3 [Rathore et al., 2024] to assess the toxicity of a given protein.

We demonstrate model performance on a modified alpha-conotoxin AuIB [Dutton et al., 2002]. The key component of this protein responsible for toxicity is the presence of the disulfide bridge. As such, we set the non-disulfide bridge components of the protein as set motifs, and then use the base-model and Safe-Genie-$5/10$ to generate residues in the place of the bridge. A model that generates toxic sequences is expected to keep the disulfide-bridge structure intact, one that does not is expected to remove the bridge. We aim to generalize this approach of targeted toxic sequence fill-ins into a broader benchmark covering a more diverse set of proteins in future work.

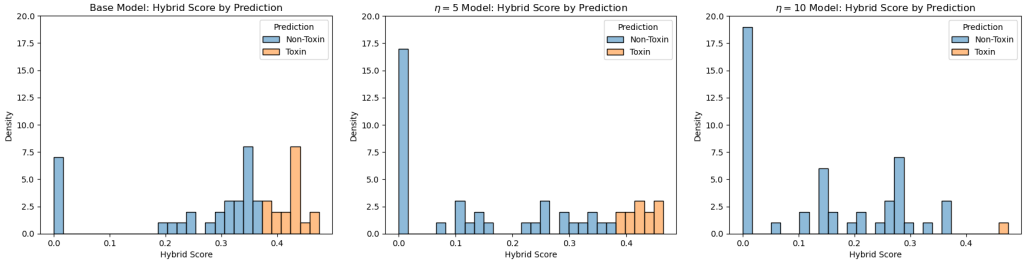


Figure 5: Comparison of ToxinPred3 score distributions and predictions for parameters 0, 5, and 10.

Figure 5 illustrates how the erasure parameter η systematically modulates the presence of the toxic concept within ToxinPred3's hybrid score distribution. At $\eta = 0$, the model cleanly separates toxic peptides (orange) above the 0.38 threshold from non-toxic peptides (blue), reflecting the baseline

learned representation of toxicity. As η increases to 5, we observe a marked compression of the non-toxic distribution toward zero, while the toxic distribution persists above threshold, albeit with reduced density near the decision boundary—consistent with partial erasure of the toxic concept while preserving its detectability. By $\eta = 10$, this erasure is nearly complete: toxic predictions are strongly attenuated, with the score distribution dominated by near-zero non-toxic instances and only faint residual traces of toxicity.

5 Looking Forwards

5.1 Limitations and Future Works

Erosion of Non-target Capabilities. A persistent limitation of concept removal is *erosion of non-target capabilities*—unintended degradation outside the targeted concept. In our protein setting, we observe a phenomenon for helix erasure: increasing the erasure strength (η) eliminates α-helices but also induces broader distributional shifts in non-target structural features (Sec. 3.1; Fig. 3). This illustrates the trade-off between safety and generative fidelity at high η. For other traits (e.g., prion-like domains), potential spillover into structurally related but benign patterns (such as ordinary β-sheets) remains a risk to evaluate empirically. Previous works have introduced methods to benchmarks to evaluate this designability-diversity tradeoff [Lin et al., 2024], in future works we plan to run these benchmarks on the various iterations of SAFEGENIE to understand how erasure impacts the broader distributions. Future avenues to explore with regards to broader distribution shifts include more localized edits (e.g., orthogonality-constrained or lightweight erasers) and multi-concept procedures designed to reduce interference [Huang et al., 2024, Gandikota et al., 2024], as well as *preservation sets* that explicitly protect benign secondary-structure distributions during editing.

Unified Toxic Protein Benchmarks. Current evaluations rely on a mixture of task-specific metrics (e.g., mPAPA for prions, ToxinPred3 for peptides) and structural analyses (e.g., helix distributions). While these provide valuable evidence of safety gains, they do not yet constitute a standardized benchmark for toxic concept erasure in generative protein models. Moreover, each was generated with ad-hoc editing of specific proteins, rather than a sustained test suite. Establishing unified benchmarks—covering toxins, prions, membrane disrupters, and other classes of unsafe proteins—is essential to enable systematic comparison across erasure algorithms and models. We envision adopting the same procedure described in Section 4, wherein toxic sub-residues are removed and diffusion models are employed to regenerate residues that fill the resulting gaps. The toxicity of the reconstructed proteins is then evaluated. Such benchmarks should balance safety evaluation with standard design metrics (designability, diversity, F1), ensuring that models are both safe and generatively useful.

Cross-Model Generalization. Our study primarily focuses on erasure in Genie-style SE(3)-equivariant diffusion models. However, the broader landscape of generative protein design includes alternative architectures such as flow-matching models, autoregressive transformers, and hybrid sequence-structure generators. Future work should investigate whether our erasure objective generalizes across these architectures and whether similar trade-offs between safety and generative fidelity emerge. Running SAFEGENIE-style algorithms on diverse model classes will help determine the robustness of erasure strategies and reveal whether unified erasure methods can provide consistent safety guarantees across the ecosystem of generative biology models.

5.2 Conclusion

We introduced SAFEGENIE, both as an algorithm and as a unified model for safe protein design. As an algorithm, SAFEGENIE provides the first parameter-level framework for erasing dangerous biological concepts from protein diffusion models, enabling tunable suppression of structural motifs (α-helices), pathogenic domains (prions), and toxin-related residues. As a model, our unified SAFEGENIE variant extends this framework by erasing thousands of toxic proteins simultaneously, yielding a single generator that balances safety with designability across diverse protein classes. This dual contribution highlights a key insight: concept erasure is not merely a defense mechanism, but a constructive tool for shaping generative distributions toward safe and useful regions of protein space. By unifying algorithmic erasure with a deployable safe model, we take a step toward standardized toxic-protein benchmarks and cross-architecture generalization, moving the field closer to generative biology models that are both powerful and responsibly deployable.

References

Abulikemu Abuduweili, Chenyang Yuan, Changliu Liu, and Frank Permenter. Enhancing sample generation of diffusion models using noise level correction. *arXiv preprint arXiv:2412.05488*, 2024. doi: 10.48550/arXiv.2412.05488.

Helen M Berman, John Westbrook, Zukang Feng, Gary Gilliland, T N Bhat, Helge Weissig, Ilya N Shindyalov, and Philip E Bourne. The protein data bank. *Nucleic Acids Research*, 28(1):235–242, 2000. doi: 10.1093/nar/28.1.235.

L. Calzolai, D. A. Lysek, P. Guntert, C. von Schroetter, R. Riek, R. Zahn, and K. Wüthrich. Nmr structures of three single-residue variants of the human prion protein. *Proceedings of the National Academy of Sciences*, 97(15):8340–8345, 2000. doi: 10.1073/pnas.97.15.8340.

Sean M. Cascarina and Eric D. Ross. Natural and pathogenic protein sequence variation affecting prion-like domains within and across human proteomes. *BMC Genomics*, 21(1):23, 2020. doi: 10.1186/s12864-019-6425-3.

David W Colby and Stanley B Prusiner. Prions. *Cold Spring Harbor Perspectives in Biology*, 3(1): a006833, 2011. doi: 10.1101/cshperspect.a006833.

Justas Dauparas, Ivan Anishchenko, Nathaniel Bennett, Hua Bai, Robert J Ragotte, Lukas F Milles, Basile I M Wicky, Adrien Courbet, Rianne J de Haas, Neville Bethel, et al. Robust deep learning-based protein sequence design using proteinmpnn. *Science*, 378(6615):49–56, 2022. doi: 10.1126/science.add2187.

J. L. Dutton, P. S. Bansal, R. C. Hogg, D. J. Adams, P. F. Alewood, and D. J. Craik. A new level of conotoxin diversity, a non-native disulfide bond connectivity in α-conotoxin auib reduces structural definition but increases biological activity. *Journal of Biological Chemistry*, 277(51):48849–48857, 2002. doi: 10.1074/jbc.M208842200. Primary citation of related structures: 1MXN, 1MXP.

Bradley Efron. Tweedie's formula and selection bias. *Journal of the American Statistical Association*, 106(496):1602–1614, 2011. doi: 10.1198/jasa.2011.tm11181.

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.

Jonathan Feldman and Tal Feldman. Resilient biosecurity in the era of ai-enabled bioweapons. *arXiv preprint arXiv:2509.02610*, 2025.

Rohit Gandikota, Joanna Materzynska, Jaden Fiotto-Kaufman, and David Bau. Erasing concepts from diffusion models. *arXiv preprint arXiv:2303.07345*, 2023. doi: 10.48550/arXiv.2303.07345. URL `https://doi.org/10.48550/arXiv.2303.07345`.

Rohit Gandikota, Hadas Orgad, Yonatan Belinkov, Joanna Materzyńska, and David Bau. Unified concept editing in diffusion models, 2024. URL `https://arxiv.org/abs/2308.14761`.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *arXiv preprint arXiv:2006.11239*, 2020. doi: 10.48550/arXiv.2006.11239.

Chi-Pin Huang, Kai-Po Chang, Chung-Ting Tsai, Yung-Hsuan Lai, Fu-En Yang, and Yu-Chiang Frank Wang. Receler: Reliable concept erasing of text-to-image diffusion models via lightweight erasers, 2024. URL `https://arxiv.org/abs/2311.17717`.

P. Hunter et al. Security challenges by ai-assisted protein design: The ability to design proteins in silico could pose a new threat for biosecurity. *EMBO Reports*, 2024. doi: 10.1038/s44319-024-00124-7.

John B Ingraham, Max Baranov, Zak Costello, Karl W Barber, Wujie Wang, Ahmed Ismail, Vincent Frappier, Dana M Lord, Christopher Ng-Thow-Hing, Erik R Van Vlack, Shan Tie, Vincent Xue, Sarah C Cowles, Alan Leung, João V Rodrigues, Claudio L Morales-Perez, Alex M Ayoub, Robin Green, Katherine Puentes, Frank Oplinger, Nishant V Panwar, Fritz Obermeyer, Adam R Root, Andrew L Beam, Frank J Poelwijk, and Gevorg Grigoryan. Illuminating protein space with a programmable generative model. *Nature*, 623(7989):1070–1078, November 2023. doi: 10.1038/s41586-023-06728-8.

Bowen Jing, Bonnie Berger, and Tommi Jaakkola. Alphafold meets flow matching for generating protein ensembles, 2024. URL https://arxiv.org/abs/2402.04845.

Mohamed Amine Ketata, Cedrik Laue, Ruslan Mammadov, Hannes Stärk, Menghua Wu, Gabriele Corso, Céline Marquet, Regina Barzilay, and Tommi S. Jaakkola. Diffdock-pp: Rigid protein-protein docking with diffusion models, 2023. URL https://arxiv.org/abs/2304.03889.

Yeqing Lin and Mohammed AlQuraishi. Generating novel, designable, and diverse protein structures by equivariantly diffusing oriented residue clouds. *arXiv preprint arXiv:2301.12485*, 2023. doi: 10.48550/arXiv.2301.12485. URL https://doi.org/10.48550/arXiv.2301.12485.

Yeqing Lin, Minji Lee, Zhao Zhang, and Mohammed AlQuraishi. Out of many, one: Designing and scaffolding proteins at the scale of the structural universe with genie 2. *arXiv preprint arXiv:2405.15489*, 2024. URL https://doi.org/10.48550/arXiv.2405.15489. Submitted on 24 May 2024.

Yikai Liu, Abhilash Sahoo, Zongxin Yu, Guang Lin, Ming Chen, and Sonya M. Hanson. Egdiff: An experiment-guided diffusion model for protein conformational ensemble generation. *bioRxiv*, 2024. doi: 10.1101/2024.10.04.616517. URL https://www.biorxiv.org/content/early/2024/10/04/2024.10.04.616517.

Jaspreet Pannu, Doni Bloomfield, Robert MacKnight, Moritz S. Hanke, Alex Zhu, Gabe Gomes, Anita Cicero, and Thomas V. Inglesby. Dual-use capabilities of concern of biological ai models. *PLoS Computational Biology*, 21(9):e1012975, 2025. doi: 10.1371/journal.pcbi.1012975. URL https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1012975.

Marc Perea, Ivar Lugtenburg, Eduardo Mayol, Arnau Cordomí, et al. TMalphaDB and TMbetaDB: Web servers to study the structural role of sequence motifs in α-helix and β-barrel domains of membrane proteins. *BMC Bioinformatics*, 16(1):266, Aug 2015. doi: 10.1186/s12859-015-0699-5.

Abhishek Singh Rathore, Ankit Arora, Suryoday Choudhury, Pranay Tijare, and Gajendra PS Raghava. Toxinpred3.0: An improved method for predicting the toxicity of peptides. *Computers in Biology and Medicine*, 179:108926, 2024. doi: 10.1016/j.compbiomed.2024.108926. URL https://doi.org/10.1016/j.compbiomed.2024.108926.

James A. Toombs, Benjamin R. McCarty, and Eric D. Ross. Compositional determinants of prion formation in yeast. *Molecular and Cellular Biology*, 30(1):319–332, 2010. doi: 10.1128/MCB.01140-09.

Juergens D. Bennett N.R. et al. Watson, J.L. De novo design of protein structure and function with rfdiffusion. *Nature*, 620:1089–1100, 2023. doi: 10.1038/s41586-023-06415-8. URL https://doi.org/10.1038/s41586-023-06415-8.

Jason Yim, Brian L. Trippe, Valentin De Bortoli, Emile Mathieu, Arnaud Doucet, Regina Barzilay, and Tommi Jaakkola. Se(3) diffusion model with application to protein backbone generation, 2023. URL https://arxiv.org/abs/2302.02277.

Vinicius Zambaldi, David La, Alexander E. Chu, Harshnira Patani, Amy E. Danson, Tristan O. C. Kwan, Thomas Frerix, Rosalia G. Schneider, David Saxton, Ashok Thillaisundaram, Zachary Wu, Isabel Moraes, Oskar Lange, Eliseo Papa, Gabriella Stanton, Victor Martin, Sukhdeep Singh, Lai H. Wong, Russ Bates, Simon A. Kohl, Josh Abramson, Andrew W. Senior, Yilmaz Alguel, Mary Y. Wu, Irene M. Aspalter, Katie Bentley, David L. V. Bauer, Peter Cherepanov, Demis Hassabis, Pushmeet Kohli, Rob Fergus, and Jue Wang. De novo design of high-affinity protein binders with alphaproteo. *arXiv preprint arXiv:2409.08022*, 2024. URL https://doi.org/10.48550/arXiv.2409.08022. 45 pages, 17 figures.

Zaixi Zhang, Ruofan Jin, Kaidi Fu, Le Cong, Marinka Zitnik, and Mengdi Wang. Foldmark: Protecting protein generative models with watermarking. *arXiv preprint arXiv:2410.20354*, 2024. Proposes watermarking for protein generative models, including diffusion architectures like RFDiffusion and FoldFlow.

A Erasure Objective Derivativation

We derive

$$\epsilon_\theta(x_t, t) \;=\; \epsilon_{\theta^*}(x_t, t) \;-\; \eta\big(\epsilon_{\theta^*}(x_t, c, t) - \epsilon_{\theta^*}(x_t, t)\big), \tag{6}$$

starting from a power–law reweighting that down-weights concept c.

Let $p_{\theta^*}(x)$ be the base data distribution and let c denote an (undesired) concept. Define the reweighted distribution

$$p_\theta(x) \;\propto\; \frac{p_{\theta^*}(x)}{p_{\theta^*}(c \mid x)^\eta}, \qquad \eta \geq 0. \tag{7}$$

Let $q_t(x_t \mid x_0)$ be the forward diffusion kernel and $p_{\theta^*}(x_t) = \int q_t(x_t \mid x_0) p_{\theta^*}(x_0)\, dx_0$, $p_{\theta^*}(x_t \mid c) = \int q_t(x_t \mid x_0) p_{\theta^*}(x_0 \mid c)\, dx_0$. Write the time-$t$ *scores*:

$$s_{\theta^*}(x_t, t) \;=\; \nabla_{x_t} \log p_{\theta^*}(x_t), \qquad s_{\theta^*}(x_t, c, t) \;=\; \nabla_{x_t} \log p_{\theta^*}(x_t \mid c). \tag{8}$$

By Bayes' rule,

$$\log p_{\theta^*}(c \mid x_t) \;=\; \log p_{\theta^*}(x_t \mid c) + \log p_{\theta^*}(c) - \log p_{\theta^*}(x_t), \tag{9}$$

so differentiating w.r.t. x_t gives

$$\nabla_{x_t} \log p_{\theta^*}(c \mid x_t) \;=\; \nabla_{x_t} \log p_{\theta^*}(x_t \mid c) - \nabla_{x_t} \log p_{\theta^*}(x_t) \;=\; s_{\theta^*}(x_t, c, t) - s_{\theta^*}(x_t, t). \tag{10}$$

From (7),

$$\log p_\theta(x_t) = \log p_{\theta^*}(x_t) - \eta \log p_{\theta^*}(c \mid x_t) + \text{const}, \tag{11}$$

$$\Rightarrow \quad s_\theta(x_t, t) = \nabla_{x_t} \log p_\theta(x_t) = s_{\theta^*}(x_t, t) - \eta\, \nabla_{x_t} \log p_{\theta^*}(c \mid x_t). \tag{12}$$

Using (10),

$$s_\theta(x_t, t) = s_{\theta^*}(x_t, t) - \eta\big(s_{\theta^*}(x_t, c, t) - s_{\theta^*}(x_t, t)\big). \tag{13}$$

For Gaussian forward noising, Tweedie's formula [Efron, 2011] yields a linear (time-dependent) map between the score and the denoiser's noise prediction:

$$\epsilon_\theta(x_t, t) \;=\; \mathcal{A}_t\, s_\theta(x_t, t), \qquad \epsilon_{\theta^*}(x_t, \cdot, t) \;=\; \mathcal{A}_t\, s_{\theta^*}(x_t, \cdot, t), \tag{14}$$

where $\mathcal{A}_t$ is the same linear operator for all conditionings at fixed t (it depends only on the diffusion schedule). Applying (14) to (13) and using linearity of $\mathcal{A}_t$,

$$\epsilon_\theta(x_t, t) = \mathcal{A}_t s_\theta(x_t, t) = \mathcal{A}_t\big[s_{\theta^*}(x_t, t) - \eta\big(s_{\theta^*}(x_t, c, t) - s_{\theta^*}(x_t, t)\big)\big] \tag{15}$$

$$= \epsilon_{\theta^*}(x_t, t) - \eta\big(\epsilon_{\theta^*}(x_t, c, t) - \epsilon_{\theta^*}(x_t, t)\big), \tag{16}$$

which is (6).

The difference $\epsilon_{\theta^*}(x_t, c, t) - \epsilon_{\theta^*}(x_t, t)$ isolates the concept-c direction at time t; subtracting η times this component removes the concept with tunable strength while preserving non-c content. (xt

B Ablation Studies and Interpretability

To understand the role of different layers in the Genie2 denoiser, we individually fine-tune the weights of specific sub-modules while keeping the remaining weights fixed. More specifically, we repeat the Alpha Helix Erasure experiment described in Section 3.1 with $\eta = 2$, while fine-tuning **only** $\theta_{SFN}, \theta_{PFN}$, or θ_{PTN}. Information on experimental procedure and hyper-parameter information can be found in Appendix C.

Figure 6 shows the loss curves from fine-tuning different weights. The "Single Feature" (blue) model converges slowly and exhibits high variance across steps, suggesting that residue-level encodings alone are insufficient to capture structural constraints. In contrast, the "Pair Feature" (green) and "Transformer" (orange) ablations both achieve more stable convergence, with the transformer-driven updates yielding particularly rapid and consistent decreases in loss. Unsurprisingly, fine-tuning *all* layers (red) leads to the lowest final loss, demonstrating the complementary nature of these modules.

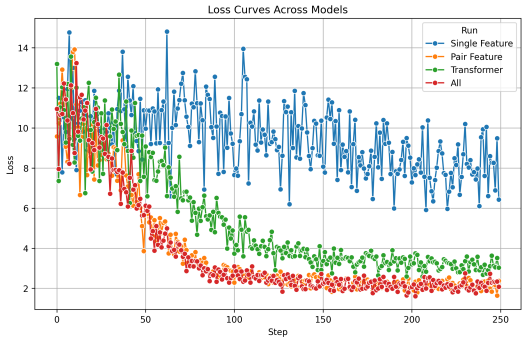


Figure 6: Loss Curve with Different Layers Unfrozen

The shape of the curves can be explained by the representational roles of each component. Single-feature embeddings primarily encode residue-level information such as type, position, and chain identity, but without pairwise or structural context, they cannot easily adapt to erasure tasks. Pairwise features incorporate inter-residue distances and orientations, allowing the model to more directly compensate for missing structural information, hence their stronger performance. The transformer-style triangular updates refine pair encodings through higher-order attention, further stabilizing training. The combined optimization shows that the modules interact synergistically rather than redundantly.

We hypothesize that the differences in convergence reflect the inductive biases each layer provides. The single feature network constrains learning to local residue identities, while the pair feature and transformer layers introduce global geometric reasoning. Thus, the improvement from adding each layer indicates that Genie2 distributes structural knowledge across levels of abstraction.

From an interpretability standpoint, these ablations clarify how structural constraints are encoded in the model. The pair and transformer modules are most directly responsible for enforcing geometric consistency, whereas the single feature network mainly anchors residue identities. This division of labor suggests that future interpretability analyses should focus on the pairwise and triangular update mechanisms when probing how Genie2 encodes motif-level or global structural information.

C Alpha Helix Erasure

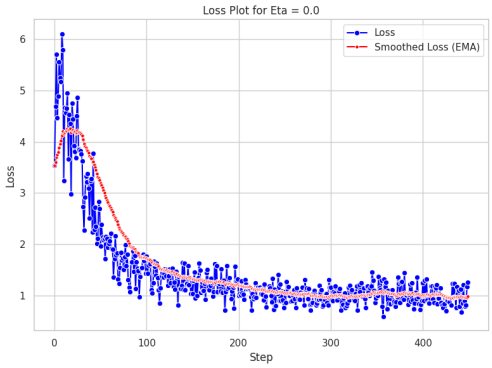
We select the follow 14 proteins from the TMalphaDB database: 2oar, 3am6, 4bem, 4fbz, 4hyj, 4pop, 4qnc, 4qnd, 4rng, 4tsy, 4wab, 4wav, 4xu4, 5ax0, and 5cbg. For each protein, we extract the alpha-helix motif's associated with chain A, as specified by the Protein Data Bank file [Berman et al., 2000]. We treat these motifs as concepts to condition on by selecting them as motif's in the Motif Scaffolding Problem Definition File and then artificially generate scaffold around the motif's to reach a protein length of 50 to 256. Recall the artificial scaffold does not matter for the loss function, as we only use the motif to calculate the loss.

We fine tune the model on values of $\eta \in \{0, 0.5, 1, 10\}$ with a learning rate of $2 * 10^{-5}$, a warmup of 50, 2 samples per step, a max-gradient norm of 40, 300 steps, and gradient accumulation every 8 steps. Loss plots can be seen in Fig 7.

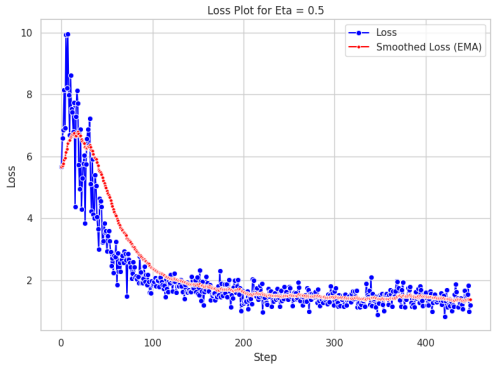
D Prion Erasure

We select the follow 4 human prion proteins: 1e1p, 1e1s, 1e1u, and 1e1w [Calzolai et al., 2000]. We treat the entire protein as a motif, or concepts to condition on, and then fine tune on the protein.

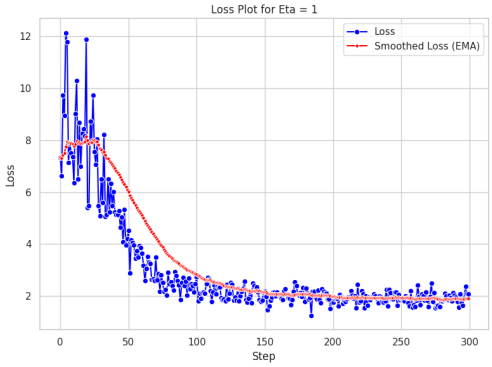
We fine tune the model on values of $\eta \in \{5, 10\}$ with a learning rate of $2 * 10^{-5}$, a warmup of 50, 2 samples per step, a max-gradient norm of 40, 300 steps, and gradient accumulation every 8 steps. Loss plots can be seen in Fig 8.



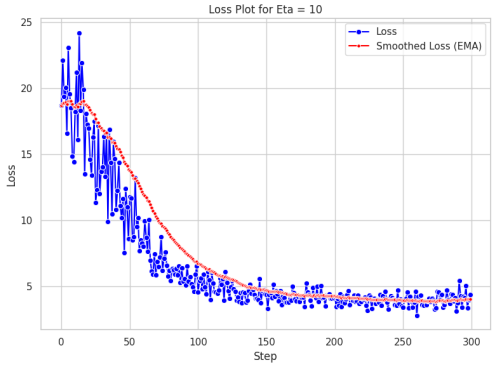
(a) Loss curve, $\eta = 0$



(b) Loss curve, $\eta = 0.5$

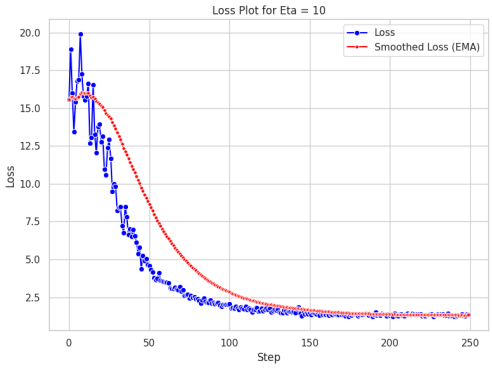


(c) Loss curve, $\eta = 1$

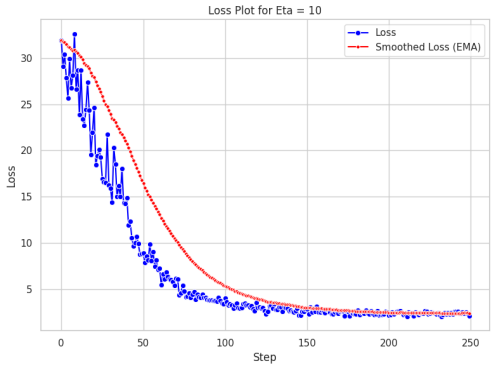


(d) Loss curve, $\eta = 10$

Figure 7: Training loss curves for different values of η for Alpha Helices.



(a) $\eta = 5$



(b) $\eta = 10$

Figure 8: Training loss curves for different values of η for Prions.

E Diffusion Sampling Process

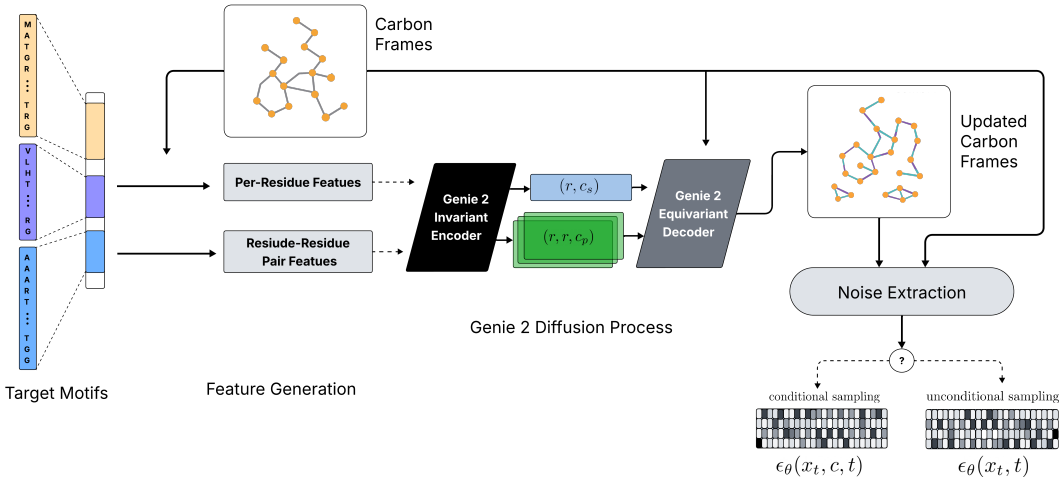


Figure 9: Conditional and Unconditional Sampling Pipeline

Genie follows a DDPM framework that generates protein backbones as sequences of C_α coordinates, performing diffusion directly in Cartesian space. At each step, an SE(3)-equivariant denoiser predicts noise displacements by reasoning over residue frames, with a single-feature encoder (residue embeddings), a pair-feature encoder (distance/orientation features), and a transformer-style pair update block that enforces global geometric consistency. Figure 9 shows this SE(3)-equivariant architecture, which we use for both inference and fine-tuning. In SAFEGENIE, we fine-tune the same denoiser parameters θ under our erasure objective (Eqs. (4)–(5)); when sampling with the edited model, the targeted motifs (e.g., α-helices or prion-like domains) are suppressed in generated structures (see Sec. 3.1).

F Unified Model Training

Preprocessing. We convert raw PDBs to a uniform, backbone-only format expected by our training pipeline. The script (i) retains only C_α atoms, (ii) selects the highest-occupancy C_α per residue index, (iii) normalizes common nonstandard residue names to canonical 3-letter codes, (iv) renumbers residues sequentially as $1..L$ in a single chain A and renumbers atom serials accordingly, and (v) writes fixed-width PDB-style lines together with a compact REMARK header. Structures with $L > 255$ are skipped.

Residue normalization. We map frequent nonstandard/modified residue codes to their canonical counterparts to avoid downstream tokenization or feature issues:

Nonstandard → Canonical	Examples
SEC → CYS, MSE → MET	seleno variants
HSD/HSE/HSP → HIS	histidine protonation/tautomer codes
GLX → GLU, ASX → ASP	ambiguous GLN/GLU and ASN/ASP
CSO/CSE/CSD → CYS	oxidized cysteine variants
SEP → SER, TPO → THR, PTR → TYR	phosphorylated residues

Unlisted residue names pass through unchanged.

Selection and renumbering. For each PDB residue index (column 23–26), we keep the C_α atom with the highest occupancy; ties are resolved by first occurrence in the file. Residues are then sorted by their original indices and reassigned consecutive IDs ($1..L$) in chain A. Note that residues are keyed only by the original residue index; if multiple chains share the same index, they are collapsed into a single sequence.

Output format. Each processed file begins with:

```
REMARK 999 NAME   <basename>
REMARK 999 PDB    <basename>
REMARK 999 INPUT A   1 <L> A
REMARK 999 MINIMUM TOTAL LENGTH      <L>
REMARK 999 MAXIMUM TOTAL LENGTH      <L>
```

followed by one ATOM line per residue (chain A, new residue IDs $1..L$) with preserved x, y, z coordinates and the occupancy/B-factor parsed from the original line, and finally `END`. Files with $L > 255$ are skipped with a console message.

Rationale and compatibility notes.

- **Equivariance features.** Genie/SAFEGENIE build pairwise geometric features from C_α coordinates; ensuring *one* C_α per residue avoids ambiguity in frame construction.
- **Chain and indexing.** Unifying to chain A and sequential indices simplifies motif masks and batching (no PDB insertion codes or gaps to resolve during training).
- **Length cap.** The 255-residue limit matches our training window sizes and GPU memory profile (see Sec. C for lengths used per task).
- **Compute Resources.** Experiments were conducted on an NVIDIA A100 with 40 GB of VRAM. Experiments each took no longer than a few hours on GPUs.

Failure modes and logging. Files with no C_α records, malformed numeric fields, or $L > 255$ are skipped with a reasoned log (e.g., "too long").

Reproducibility. Processing is deterministic given an input PDB. We release the mapping table and preprocessing code to enable regeneration of the dataset.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The claims about how we can target specific traits and suppress them in the model are supported by the experimental results of the paper.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: The paper includes a dedicated section for limitations including but not limited to limitations of methods and experiments.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

d: [Yes]

Justification: The paper includes formulas throughout the main sections, as well as supplementary derivations in the appendix for the erasure formulation and training methods.

Guidelines:

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

4. **Experimental result reproducibility**

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

Answer: [Yes]

Justification: The paper includes a comprehensive description of how traits were targetted and how to setup the files in order to replicate the experiments. Code will be made available to further aid in enabling reproducibility, although the paper describes methods to an extent needed to reproduce experiments.

Guidelines:

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

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

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

   Answer: [Yes]

   Justification: Yes, code provided and hosted on github with weights provided. Additional weights from referenced papers are also accessable through the URLs in the references section.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: Yes, experiments are carefully outlined and the appendix provides training details so that data can be preprocessed in a deterministic manner. Training methods are also carefully described for each case study.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: The paper reports plots with exact experimental values, so many results do not require error bars. However, derivations and methods are provided carefully as so to allow one to understand the statistical significance of the experiments.

   Guidelines:

   - The answer NA means that the paper does not include experiments.

- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: Compute specifications are identical to those of the paper [Lin et al., 2024], however we provide exact compute resources we used.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: We reviewed and follow all code of conduct guidelines.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

Justification: We situate the work in the broader context of biosafety regulation and discuss how erasure could be both used for positive (toxic and prion) erasure but also degrade model quality (in the case of alpha helices).

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

Answer: [NA] .

Justification: The paper poses no such risks.

Guidelines:

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

12. **Licenses for existing assets**

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

Answer: [Yes]

Justification: All code is our own, or cited.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.

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

13. **New assets**

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

    Answer: [Yes]

    Justification: Code and datasets will be released after double-blind submission. Training information is all disclosed.

    Guidelines:

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

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

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

    Answer: [NA] .

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

    Guidelines:

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

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

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

    Answer: [NA]

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

    Guidelines:

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

16. **Declaration of LLM usage**

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

    Answer: [NA]

    Justification: Core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

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