AURORA: Alignment-Guided Mutation Proposal for Protein Engineering

Katie Spivakovsky [1 2 3] **Jeannie She** [1 2] **Nathan P. Shapiro** [1 3] **Krithik Ramesh** [3]

Abstract

Protein engineering introduces mutations to enhance protein function and has immense therapeutic, agricultural, and industrial applications, but experimental validation is expensive, limiting available data. The prevailing computational approach uses a protein foundation model for two tasks: an oracle built on its representations scores mutation effects, and a search procedure proposes mutations through reinforcement learning (RL). These approaches predominantly rely on single-sequence models, namely ESM, that predict masked amino acids, requiring them to infer mutation interactions from single sequences rather than mutation co-evolution patterns across multiple homologous sequences. We introduce an alignment-guided mutation proposer and oracle (AURORA), a protein engineering framework with two key components. First, we investigate the natural transition to multiple sequence alignment (MSA)-based models, namely MSA Pairformer, which directly compares homologs; we quantify architectural expressivity on synthetic proteins and find Pairformer performs better on downstream benchmarks, notably ProteinGym. Second, because representation models capture evolutionary distributions while search optimizes experimental rewards, we decouple these tasks: Pairformer scores mutants while a separate lightweight policy trained with RL proposes mutations, enabling direct multi-site proposals rather than iterative single-site search. We validate AURORA on green fluorescent protein, training on limited data to generate novel variants that demonstrate higher fluorescence than existing methods.

[1]Department of Biological Engineering, Massachusetts Institute of Technology [2]Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology [3]Lyra Labs. Correspondence to: Katie Spivakovsky <kspiv@mit.edu>, Krithik Ramesh <krithik@lyralabs.ai>.

Accepted at the 2026 Workshop on Generative and Agentic AI for Biology (ICML 2026)

1. Introduction

Proteins are biological machinery: an amino acid sequence folds into a three-dimensional structure that determines functional utility (Epstein et al., 1963). Mutations alter this function, creating variants with enhanced properties. These mutations vary in complexity: single-site substitutions change one amino acid at a time, while multi-site mutations change multiple simultaneously. Critically, mutations often interact epistatically, where one mutation's effect depends on the broader sequence context (Starr & Thornton, 2016). This creates a high-dimensional fitness landscape defined by mutation combinations. Understanding and exploiting this landscape is the central challenge of protein engineering. Wet lab directed evolution navigates this landscape through repeated cycles of mutagenesis and screening, but each cycle requires testing 10^3–10^9 variants (Cobb et al., 2013). This combinatorially exhaustive approach makes comprehensive exploration prohibitively expensive.

In contrast, machine learning-guided protein engineering uses limited experimental data to guide mutation selection (Yang et al., 2019). Across diverse tasks including antibody affinity maturation and CRISPR nuclease optimization, recent successes have required only dozens of experimentally tested variants (Hie et al., 2023; Jiang et al., 2025). However, realizing this promise requires models that can both accurately predict mutation effects from limited data and efficiently search combinatorial mutational spaces.

The difficulty arises from the rugged, highly epistatic structure of protein fitness landscapes: while single-site mutations may be beneficial in isolation, their combined effects are often non-additive (Starr & Thornton, 2016). Naive strategies that evaluate all single mutants and iteratively build on the best struggle to escape local optima and scale poorly to multi-site mutations (Wittmann et al., 2021; Johnston et al., 2024). With sparse experimental data providing limited supervision, effective protein engineering requires both representations that capture epistatic structure and search procedures that reason over mutation sequences rather than isolated substitutions.

Most existing approaches rely on pretrained protein language models (PLMs), particularly ESM, as a shared foundation for both mutation effect prediction and proposal. A supervised oracle trained on PLM embeddings predicts fitness,

then the same model generates mutations via RL, Bayesian optimization, or greedy search (Stocco et al., 2025; Cao et al., 2025; Soldát & Kléma, 2024; Benjamins et al., 2024; Hie et al., 2023; Jiang et al., 2025). This design introduces two limitations. First, single-sequence models infer epistasis indirectly from parameters rather than directly observing co-evolving mutations. Second, using one model for both representation and proposal creates architectural mismatch: representations approximate average evolutionary distributions, whereas we see mutation proposal as a decision-making problem prioritizing extraordinary behavior. Compensating often requires aggressive scaling or post-training, which does not reliably improve performance—ESM2-650M outperforms larger 3B and 15B variants on zero-shot variant effect prediction (Hou et al., 2025).

We propose a design that addresses both limitations by separating representation learning from mutation proposal. For representation, we adopt the MSA Pairformer architecture of Akiyama et al. (2025), which directly leverages multiple sequence alignments to observe co-varying mutations at inference time. For search, we formulate mutation proposal as a separate Markov decision process (MDP) and train a lightweight policy using reinforcement learning to propose multi-site mutations based on oracle feedback. This decoupled design enables fitness landscape exploration without interfering with the representation model. We validate the approach on green fluorescent protein from *Aequorea victoria* (avGFP) with limited experimental data and demonstrate improved variant discovery compared to existing methods.

Our main contributions are:

- **MSA-based representations for variant effect prediction**: We show that MSA Pairformer embeddings capture epistatic interactions more effectively than single-sequence PLMs. Through synthetic experiments and downstream benchmarks, we demonstrate superior zero-shot and supervised fitness prediction.
- **Decoupled mutation proposal via reinforcement learning**: We formulate mutation proposal as an MDP operating on a fixed fitness oracle. A separate policy network trained with RL proposes mutations, emitting position and amino acid changes simultaneously. Our ablation studies show that joint action spaces enable better reasoning about multi-site mutations than autoregressive approaches that generate variants sequentially.
- **Empirical validation on avGFP engineering**: We demonstrate our architecture identifies improved protein variants using limited experimental data, outperforming baselines relying on iterative single-site search or ESM-based representations. We performed preliminary validation of predictions through in vitro fluorescence measurements of synthesized variants.

2. Background and Preliminaries

2.1. Protein Language Models and Evolutionary Statistics

Protein language models learn statistical regularities from large sequence databases, capturing constraints that distinguish functional proteins from random polypeptides. ESM (Rives et al., 2021) and ESM-2 (Lin et al., 2023) process individual sequences through masked language modeling, predicting held-out amino acids from sequence context. Despite never receiving explicit evolutionary supervision, these models learn biologically meaningful representations: their embeddings predict secondary structure, contact maps, and mutational effects (Rives et al., 2021). Recent work has shown that reinforcement learning from experimental feedback can further align PLM outputs with functional constraints (Widatalla et al., 2024), and that PLM likelihoods correlate with antibody binding affinity (Hie et al., 2023).

However, single-sequence models face a fundamental limitation for epistasis. When predicting the effect of multi-site mutations, they must infer residue interactions entirely from parameters learned during pretraining. The model never directly observes which positions co-evolve across a protein family—it must reconstruct this information implicitly from having seen millions of unrelated sequences. This architectural constraint bounds the epistatic reasoning these models can perform.

2.2. Incorporating Evolutionary Context

A natural solution is to condition predictions on evolutionary context directly. Two paradigms have emerged: homology-aware alignment-free methods and explicit MSA-based architectures.

Alignment-free approaches. PoET (Truong Jr & Bepler, 2023) and Tranception (Notin et al., 2022) retrieve homologous sequences and condition on them without constructing an explicit alignment. PoET models the conditional probability of a target sequence $\mathbf{x}$ given a set of retrieved homologs $\mathcal{H} = \{\mathbf{h}_1, \ldots, \mathbf{h}_K\}$:

$$p(\mathbf{x}|\mathcal{H}) = \prod_{i=1}^{L} p(x_i|x_{<i}, \mathcal{H}) \tag{1}$$

where the homologs provide context through cross-attention. This approach captures evolutionary information but requires the model to infer covariation structure from the unaligned sequences—a nontrivial inductive leap. ProGen (Madani et al., 2023) similarly uses autoregressive generation conditioned on control tags, enabling property-guided sequence design but inheriting the sequential limitations of left-to-right generation.

MSA Transformer. Rather than treating homologs as unstructured context, MSA Transformer (Rao et al., 2021) operates directly on aligned sequences. Given an MSA $\mathbf{M} \in \mathcal{A}^{D \times L}$ with D sequences of length L, the model applies interleaved attention: *row attention* allows each sequence to attend across positions (capturing within-sequence dependencies), while *column attention* allows each position to attend across sequences (capturing cross-sequence covariation). This bidirectional information flow enables direct extraction of co-evolutionary signals without requiring the model to infer alignment structure.

MSA Pairformer. Pairformer architectures (Akiyama et al., 2025) extend MSA processing with explicit pairwise representations. Beyond the per-residue MSA representation $\mathbf{m} \in \mathbb{R}^{D \times L \times d}$, Pairformer maintains a pair representation $\mathbf{z} \in \mathbb{R}^{L \times L \times d_p}$ encoding relationships between all position pairs. The pair representation is initialized from relative positional encodings and updated through three mechanisms:

Outer product mean extracts pairwise statistics from the MSA:

$$\mathbf{z}_{ij} \leftarrow \mathbf{z}_{ij} + \frac{1}{D} \sum_{s=1}^{D} \mathbf{W}_L \mathbf{m}_{si} \otimes \mathbf{W}_R \mathbf{m}_{sj} \tag{2}$$

where $\mathbf{W}_L, \mathbf{W}_R$ are learned projections. This operation directly computes covariation statistics across the alignment.

Triangle multiplicative updates propagate information through the pair representation using the geometric constraint that residue relationships should be consistent: if positions i and k interact, and k and j interact, this constrains the i-j relationship. The outgoing update aggregates over intermediate positions:

$$\mathbf{z}_{ij} \leftarrow \mathbf{z}_{ij} + \sum_{k} \sigma(\mathbf{g}^a_{ik}) \mathbf{a}_{ik} \odot \sigma(\mathbf{g}^b_{kj}) \mathbf{b}_{kj} \tag{3}$$

where $\mathbf{a}, \mathbf{b}$ are projections of $\mathbf{z}$ and $\sigma(\mathbf{g})$ provides gating.

The architectural distinction is fundamental. Single-sequence models *store* evolutionary information in parameters learned during pretraining. MSA-based models *read* evolutionary information from the alignment provided at inference time. This enables Pairformer to adapt to the specific evolutionary context of each query protein.

3. Epistatic Synthetic

Before applying AURORA to real proteins, we seek to understand a fundamental question: *when and why do MSA-based architectures outperform single-sequence models?* Real protein fitness landscapes confound multiple factors—evolutionary history, structural constraints, assay noise, and unknown epistatic patterns—making it difficult to isolate the specific architectural advantages of different model classes.

We therefore construct a **synthetic epistasis benchmark** satisfying three key desiderata. First, the fitness function decomposes into interpretable components (additive, pairwise epistatic, covariation-dependent, and interaction terms), allowing us to attribute model performance to specific capabilities. Second, we can systematically vary epistatic complexity, MSA depth, and covariation signal strength to identify scaling behavior. Third, the ground-truth epistatic pairs are known, enabling evaluation of whether models learn correct residue-residue dependencies.

3.1. Mathematical Formulation

3.1.1. Fitness Landscape

Consider a protein sequence $\mathbf{x} = (x_1, \ldots, x_L)$ where $x_i \in \mathcal{V} = \{1, \ldots, V\}$ represents the amino acid at position i. Given an associated multiple sequence alignment $\mathbf{M} \in \mathcal{V}^{M \times L}$, we define fitness as a sum of four terms plus noise:

$$f(\mathbf{x}, \mathbf{M}) = f_{\text{add}} + f_{\text{epi}} + f_{\text{cov}} + f_{\text{int}} + \epsilon \tag{4}$$

$$\text{where} \quad f_{\text{add}} = \sum_{i=1}^{L} h_i(x_i) \tag{5}$$

$$\begin{aligned} f_{\text{epi}} &= \sum_{(i,j) \in \mathcal{E}} J_{ij}(x_i, x_j) \\ f_{\text{cov}} &= \sum_{(i,j) \in \mathcal{E}} w_{ij} \cdot C_{ij}(\mathbf{M}) \\ f_{\text{int}} &= \sum_{(i,j) \in \mathcal{E}} V_{ij}(x_i, x_j) \cdot C_{ij}(\mathbf{M}) \end{aligned} \tag{6}$$

The functions $h_i : \mathcal{V} \to \mathbb{R}$ represent single-site fitness effects, while $\mathcal{E} \subset \{(i,j) : 1 \leq i < j \leq L\}$ denotes the set of epistatic position pairs. Pairwise epistatic couplings are captured by $J_{ij} : \mathcal{V} \times \mathcal{V} \to \mathbb{R}$, and $C_{ij}(\mathbf{M}) \in [0, 1]$ measures the covariation strength between positions i and j computed from the MSA. The weights $w_{ij} \in \mathbb{R}$ scale the contribution of pure covariation signal, while $V_{ij} : \mathcal{V} \times \mathcal{V} \to \mathbb{R}$ captures the *interaction* between sequence identity and evolutionary covariation. Finally, $\epsilon \sim \mathcal{N}(0, \sigma^2)$ represents measurement noise.

3.1.2. Covariation Matrix

The covariation matrix $\mathbf{C}(\mathbf{M})$ is computed from the MSA as follows. Let $\mathbf{O}^{(s)} \in \{0,1\}^{L \times V}$ be the one-hot encoding of sequence s in the MSA, with centered representation $\bar{\mathbf{O}}^{(s)} = \mathbf{O}^{(s)} - \frac{1}{M} \sum_{s'=1}^{M} \mathbf{O}^{(s')}$. The covariation between positions i and j is then computed as the normalized Frobe-

nius norm of the outer product:

$$C_{ij}(\mathbf{M}) = \frac{\left\| \frac{1}{M} \sum_{s=1}^{M} \bar{\mathbf{o}}_i^{(s)} \otimes \bar{\mathbf{o}}_j^{(s)} \right\|_F}{\max_{i',j'} \|\cdot\|_F} \tag{7}$$

where $\bar{\mathbf{o}}_i^{(s)} \in \mathbb{R}^V$ is the centered one-hot vector at position i for sequence s.

3.1.3. Why Different Architectures Have Different Capabilities

The fitness decomposition in Equation 4 reveals why different architectures have fundamentally different capabilities.

Single-sequence models (ESM-style) process only $\mathbf{x}$ without access to $\mathbf{M}$. They can learn additive effects $h_i(x_i)$ directly and pairwise couplings $J_{ij}(x_i, x_j)$ through attention mechanisms. However, they *cannot* access covariation $C_{ij}(\mathbf{M})$ or the interaction term $V_{ij}(x_i, x_j) \cdot C_{ij}(\mathbf{M})$. When covariation-dependent terms dominate fitness, single-sequence models face an information-theoretic barrier.

MSA Transformer processes the full MSA through row and column attention, enabling extraction of covariation signals. However, its implicit pair representation (outer product of row features) may not efficiently capture the interaction term, which requires simultaneously representing both sequence identity and evolutionary correlation.

MSA Pairformer maintains an explicit pair representation $\mathbf{Z} \in \mathbb{R}^{L \times L \times d}$, updated bidirectionally with the MSA representation via outer product mean and triangle attention. This architecture is suited to model $Z_{ij} \approx g(x_i, x_j, C_{ij}(\mathbf{M}))$, enabling direct representation of the interaction term. The triangle multiplicative updates $Z_{ij} \leftarrow Z_{ij} + \sum_k Z_{ik} \odot Z_{kj}$ further enforce geometric consistency, propagating epistatic information through indirect paths.

3.2. Experimental Setup

We instantiate the benchmark with sequence length $L = 32$ and vocabulary size $V = 20$. The default configuration uses $|\mathcal{E}| = 20$ epistatic pairs and MSA depth $M = 64$, with fitness scales $\sigma_h = 1.0$ (additive), $\sigma_J = 2.0$ (epistatic), $\sigma_w = 3.0$ (covariation), and $\sigma_V = 1.5$ (interaction), plus noise $\sigma = 0.1$. We generate 2,000 training samples, 400 validation samples, and 500 test samples, where each variant contains 1–5 mutations from a fit wild-type sequence.

We compare three architectures with matched parameter budgets: **ESM** (single-sequence transformer with no MSA access), **MSA-Transformer** (row and column attention on MSA), and **Pairformer** (bidirectional MSA-pair updates with triangle operations).

Table 1. Synthetic benchmark summary. Spearman ρ across ablation conditions. PF = Pairformer, MSA-T = MSA-Transformer.

Ablation	Condition	ESM	MSA-T	PF
MSA Depth	Shallow (4)	.523	.739	**.750**
	Deep (128)	.692	.901	**.933**
Covariation	None (0)	.924	.925	**.935**
	Strong (5)	.533	.855	**.904**
Epistasis	Simple (5)	.925	.907	**.927**
	Complex (80)	.838	.894	**.902**
Data Size	Small (500)	.570	.665	**.700**
	Large (4000)	.727	.917	**.931**
Mutations	Single (1)	.578	.896	**.923**
	Multi (5)	.664	.814	**.844**

4. Methods

4.1. Oracle Architectures and Training

We trained six oracle models to predict avGFP fitness using a deep mutational scanning (DMS) dataset containing in vitro activity measurements for 51,744 sequences (Sarkisyan et al., 2016), excluding 2,280 variants with nonsense mutations. The six oracles resulted from combining three sequence representations—raw sequence, ESM-2 embeddings, and MSA Pairformer embeddings—with two downstream architectures: multi-layer perceptrons (MLPs) and BiMamba models.

For sequence representations, raw sequences were encoded using a Transformer encoder, ESM-2 650M embeddings (Lin et al., 2023) were generated and averaged across residue positions to produce fixed-length vectors, and MSA Pairformer embeddings were extracted from the pretrained model. The MLP architecture consisted of three fully connected layers ($1280 \to 320 \to 128 \to 1$) with ReLU activations between layers. The BiMamba architecture followed the design from literature, adapting it for protein fitness prediction.

4.2. Mutation Proposal with Reinforcement Learning

Similar to prior work (Sun et al., 2025), we formulated mutation proposal as a Markov decision process (MDP). The state $s_t \in \mathcal{A}^N$ represents the current protein sequence as a length-N vector of amino acid indices. At each timestep, the agent selects an action $a_t = (i, a)$ corresponding to a point mutation that substitutes amino acid a at residue position i. The action space is thus the Cartesian product $\{1, \ldots, N\} \times \{1, \ldots, 20\}$, encompassing all single-residue substitutions across the sequence. We set the episode horizon to $H = 6$ mutations, as prior work has shown exponential decay in fitness as mutation burden increases (Tokuriki & Tawfik, 2009). The reward signal r_t at each timestep is the oracle-predicted mutant activity. We trained policies using PPO

with the following hyperparameters: $\lambda_{\text{GAE}} = 0.95$, $n_{\text{envs}} = 64$, $n_{\text{steps}} = 6$, $\gamma = 0.99$, and total timesteps of 10,000.

5. Results

We present three main contributions that establish MSA Pairformer as a powerful architecture for protein engineering. First, in Section 5.1, we investigate the architectural expressivity of MSA Pairformer and demonstrate its superior ability to capture epistatic interactions compared to single-sequence models like ESM, both through evaluation on synthetic benchmarks and comprehensive downstream tasks: ProteinGym and BindGym. Second, in Section 5.2, we introduce a novel RL approach for proposing mutations that departs from canonical autoregressive sampling. We demonstrate that MSA Pairformer oracles encourage broader exploration of sequence space, generating mutations that are both functionally promising and distinct from the training distribution. Finally, in Section 5.3, we validate our approach through characterization of improved green fluorescent protein variants, and begin work confirming that computationally predicted mutations translate to measurable functional improvements in the laboratory.

5.1. MSA Pairformer Captures Epistatic Interactions Better than ESM

5.1.1. MSA Pairformer Oracles Achieve the Best Fitness Prediction

Oracle performance was evaluated using Spearman rank correlation and R^2 on held-out variants via 5-fold random cross-validation. MSA Pairformer embeddings paired with a BiMamba oracle achieved the best performance on both metrics ($\rho = 0.863, R^2 = 0.872$). The two metrics diverge for the weaker representations: raw-sequence oracles were competitive on rank correlation — raw/BiMamba ($\rho = 0.851$) ranked second overall and surpassed all ESM configurations — yet the MSA Pairformer advantage was most pronounced on R^2, where it explained substantially more variance (0.781–0.872) than raw or ESM inputs (0.59–0.68). Across representations, BiMamba oracles consistently outperformed MLPs, indicating improved modeling of sequence-dependent effects.

Table 2. Oracle Architecture Ablation Study

Input	Oracle	Spearman (ρ)	R^2
Raw Seq	MLP	0.843 ± 0.004	0.667 ± 0.029
Raw Seq	BiMamba	0.851 ± 0.006	0.683 ± 0.024
ESM	MLP	0.716 ± 0.003	0.592 ± 0.011
ESM	BiMamba	0.812 ± 0.008	0.654 ± 0.018
MSA PF	MLP	0.828 ± 0.008	0.781 ± 0.015
MSA PF	BiMamba	$\mathbf{0.863 \pm 0.005}$	$\mathbf{0.872 \pm 0.012}$

5.1.2. ProteinGym Benchmarking

To validate that our synthetic benchmark findings transfer to real proteins, we evaluate Pairformer against recent protein language models on 88 variant effect prediction tasks from ProteinGym (Notin et al., 2023). Table 3 presents the results. Despite using approximately 5× fewer parameters (110M vs. 600M), Pairformer achieves the highest mean Spearman correlation ($\rho = 0.710$) and R^2 (0.442), outperforming larger models including Profluent E1-600M in both single-sequence and retrieval-augmented configurations, as well as ESMC-600M. The performance gap is particularly notable for R^2, where Pairformer's 0.442 substantially exceeds the next-best result of 0.341—a 30% relative improvement. These results confirm that MSA-based architectures with explicit pair representations provide superior fitness prediction on diverse real-world proteins, validating our choice of Pairformer as the reward model backbone for AURORA.

Table 3. Variant Effect Prediction Performance (n=88 tasks)

Model	Params	Spearman (ρ)	R^2
Pairformer	111M	$\mathbf{0.710 \pm 0.158}$	$\mathbf{0.442 \pm 0.185}$
E1-600M Single	600M	0.690 ± 0.161	0.341 ± 0.189
E1-600M Retrieval	600M	0.687 ± 0.163	0.303 ± 0.221
ESMC-600M	600M	0.686 ± 0.164	0.334 ± 0.177

5.2. Novel Mutation Proposal

Following the training protocol described in Section 4.2, we trained six PPO agents, each using a different oracle as its reward function. All agents exhibited stable learning dynamics with monotonically increasing rewards. However, after approximately 10,000 timesteps, policies began to exhibit mode collapse, converging to repeatedly select specific high-reward actions. To preserve exploration diversity and prevent degenerate policies, we terminated training before reward plateaued and extracted the final policy for downstream analysis.

5.2.1. Contrasting Canonical Autoregressive

Causal models consistently underperformed bidirectional models on avGFP fitness prediction. Bidirectional Mamba2 achieved the highest performance (R^2=0.796, ρ=0.850), followed by Autoregressive Mamba2 (Dao & Gu, 2024) (R^2=0.730, ρ=0.847), while both transformer variants lagged substantially—Bidirectional Transformer (R^2=0.635, ρ=0.708) and Causal Transformer (R^2=0.584, ρ=0.714). Within each architecture family, bidirectional variants outperformed their causal counterparts, with the gap most pronounced for Mamba models where bidirectionality improved R^2 by 9%.

To understand this performance gap mechanistically, we computed categorical Jacobians (Zhang et al., 2024) measur-

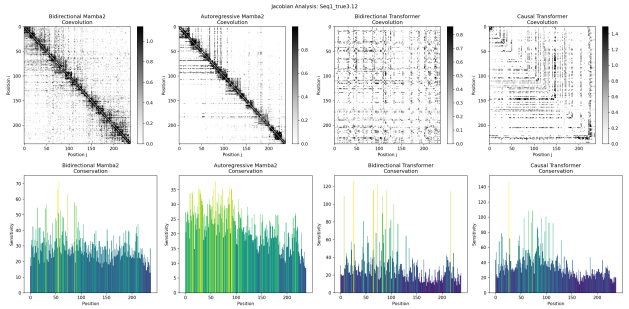

Figure 1. **Coevolution and conservation maps comparing Bidirectional Mamba2 and Transformer Variants** (a) Training with a `Pairformer / MLP` oracle.

ing how input perturbations propagate through each model. The coevolution maps reveal a stark architectural signature: bidirectional models exhibit dense off-diagonal coupling, indicating that mutations at position i influence predictions at distant positions j regardless of sequence order. Causal models, by contrast, show characteristic triangular structure, the Causal Transformer's Jacobian is nearly zero for $j > i$, reflecting the autoregressive mask that prevents positions from influencing earlier residues. This asymmetry constitutes epistatic myopia: causal models cannot capture compensatory mutations where early and late positions must co-evolve. The conservation profiles corroborate this finding, with bidirectional models showing clearer sensitivity peaks at known functional sites. These results support our decoupled design: rather than forcing an autoregressive generator to simultaneously model epistasis and propose mutations, AURORA uses bidirectional representations for the oracle while delegating generation to a separate policy that can leverage the oracle's richer understanding of epistatic landscapes.

5.2.2. MSA Pairformer Oracles Encourage Exploration of Sequence Space

Beyond absolute fitness improvements, effective protein design requires exploring diverse regions of sequence space to avoid local optima and discover non-obvious beneficial mutations. We investigated whether different oracle architectures influence the exploratory behavior of RL agents by analyzing the distribution of proposed mutations across sequence and embedding space during PPO training.

We recorded the visitation frequency of each discrete action throughout PPO optimization and aggregated these counts over the full training horizon. We then visualized the resulting empirical action distributions as heatmaps, enabling a direct comparison of which regions of the action space were explored under different oracle architectures. When using a baseline MLP policy backbone, MSA Pairformer-based training leads to substantially broader action visitation compared to ESM-based training, making the difference in exploration particularly pronounced (Fig. 2). In contrast, when using a BiMamba policy backbone, policies trained with the MSA Pairformer–Mamba oracle exhibit only a modest but consistent increase in coverage across the action space relative to those trained with the ESM–Mamba oracle, suggesting a less concentrated exploration pattern (Fig. 6 in Appendix). Together, these results indicate that oracle choice affects not only final performance but also the exploratory behavior induced during policy learning, with architectural inductive biases playing a larger role when the policy model is less expressive.

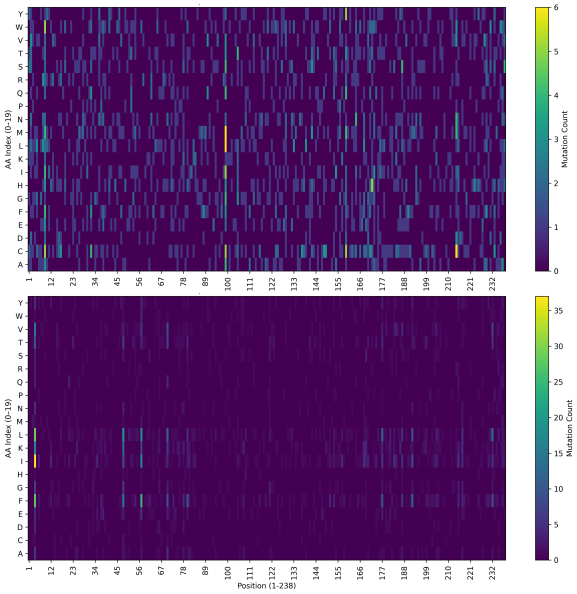

Figure 2. **Heatmaps of action space visitation frequency during PPO training.** The top panel corresponds to training with a `Pairformer / MLP` oracle, while the bottom panel corresponds to an `ESM / MLP` oracle. By visual inspection, the `Pairformer / MLP`–guided policy exhibits slightly broader exploration across the action space. Note that the color scale differs between panels, with the `Pairformer / MLP` heatmap spanning a smaller range, reflecting a more distributed visitation pattern.

5.2.3. Proposals are Novel to Dataset

To assess whether RL agents simply recapitulate the training data or explore novel regions of sequence space, we compared the mutation profiles of RL-proposed sequences against the empirical DMS dataset. We extracted all single-point mutations from both the DMS dataset and the sequences generated by our trained PPO agents, then calculated the frequency of each position-mutation pair.

The mutation landscapes differ substantially between

datasets (Fig. 3). While some positions show overlap, indicating the oracle correctly assigns high fitness to empirically validated mutations, the RL agents propose mutations at numerous positions that are rare or absent in the DMS data. Furthermore, at positions mutated in both datasets, RL agents often favor different amino acid substitutions than those observed empirically. For instance, at position 135, the DMS dataset shows predominant mutations to serine or aspartic acid, while RL agents propose substitutions to phenylalanine with comparable frequency.

These divergent mutation profiles demonstrate that RL agents are not merely memorizing high-fitness variants from the training distribution. Instead, they leverage the oracle's learned fitness landscape to propose combinations of mutations that explore underrepresented regions of sequence space, potentially identifying beneficial variants that were not sampled in the original experimental library.

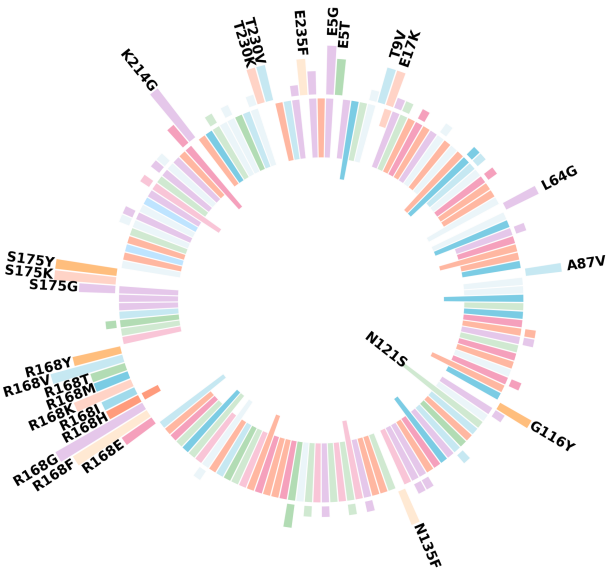

Figure 3. **RL-proposed mutations are novel compared to DMS dataset.** Comparison of mutation frequencies between the DMS (inner ring) and mutations proposed by a representative RL model (outer ring), namely the model based on the `RAW SEQ / MLP` oracle. Each bar represents a single-point mutation. Bar height indicates mutation frequency as a percentage of total mutations in each dataset, with colors denoting amino acid identity. Mutations appearing at $\geq$0.3% frequency in at least one dataset are shown.

5.3. Validation of Improved Green Fluorescent Protein

To assess whether improved oracle modeling translates to experimentally measurable gains, we designed novel avGFP mutants through our approach and quantified their fluorescence in the wet lab.

We trained six oracles according to the methods earlier described. We then used these oracles as the reward signal for our RL-based mutant proposal method earlier described. After training each policy, we sampled to generate two double–, two quadruple–, two quintuple–, and one sextuple–mutant. Altogether, we generated 42 mutants, and confirmed all were novel to the DMS dataset.

Using the conservative `RAW SEQ / pairformer MLP` oracle, we evaluated the predicted fitness of our 42 RL-generated mutants against the full DMS dataset of 51,744 sequences. RL mutants exhibited substantially higher predicted fitness (mean = 3.56, SD = 0.84) compared to the DMS distribution (mean = 2.66, SD = 1.06), with a large effect size (Cohen's d = 0.86). This difference was highly significant across multiple statistical tests: two-sample t-test ($t = 5.54$, $p < 0.0001$), Mann-Whitney U test ($p < 0.0001$), and Kolmogorov-Smirnov test (KS = 0.76, $p < 0.0001$). Notably, 36/42 (86%) of RL mutants exceeded the median DMS fitness, 34/42 (81%) exceeded the 75th percentile, and 33/42 (79%) exceeded the 90th percentile. Furthermore, 28/42 (67%) of RL-generated sequences surpassed the 99th percentile of the DMS dataset, demonstrating that reinforcement learning successfully enriched for high-fitness variants beyond those well-represented in the training data (Fig. 4).

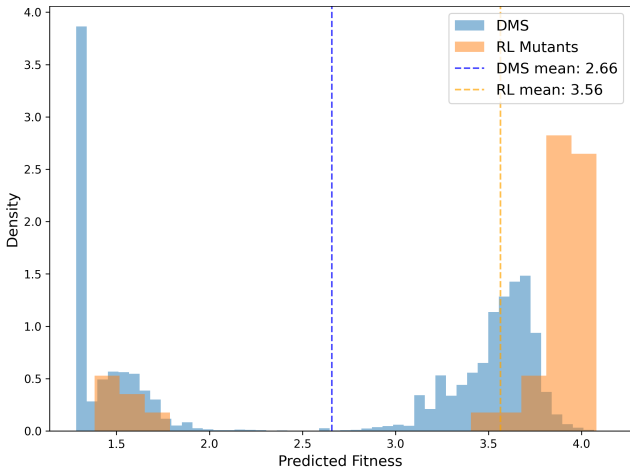

Figure 4. **Distribution of predicted fitness scores for DMS and RL-generated mutants.** We compare the predicted fitness of deep mutational scanning (DMS; blue) variants and reinforcement-learning–designed (RL; orange) mutants through histograms. Vertical dashed lines denote the mean predicted fitness for each set (DMS: 2.66; RL: 3.56). RL mutants are shifted toward higher predicted fitness values relative to DMS variants, indicating successful enrichment for high-fitness sequences.

We proceeded to experimentally validate all 42 mutants. For each mutant, we synthesized DNA encoding a fusion protein linking the avGFP sequence to mKate protein via a rigid alpha-helix linker, enabling rigorous quantification across

conditions by normalizing to the mKate signal. Constructs were expressed in duplicate and fluorescence was measured by plate reader.

Raw avGFP and mKate fluorescence values were background-corrected by subtracting the median signal of blank wells. To guard against artifacts from poorly expressing constructs, wells with background-corrected mKate signal below 10% of the wild-type median were excluded from all downstream analysis.

Relative fluorescence activity for each variant was estimated by fitting a linear model of the form `avGFP=` $\alpha\cdot$`mKate` through the origin. This slope was estimated separately for wild-type (α_{WT}) and each variant (α_{var}), and relative activity was defined as $\alpha_{\text{var}}/\alpha_{\text{WT}}$. Uncertainty in the relative activity was propagated from the standard errors of both slopes via the delta method.

Statistical significance was assessed by applying Welch's two-sample t-test to log-transformed GFP/mKate ratios, comparing each variant against wild-type replicates. P-values were corrected using Benjamini–Hochberg false discovery rate, with significance called at $q < 0.05$.

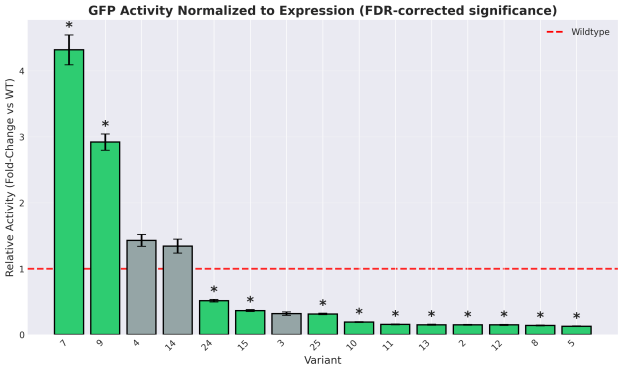

Figure 5. **Fluorescence of top performing mutants, normalized to wild-type expression.** Bar height indicates the activity ratio $\alpha_{\text{var}}/\alpha_{\text{WT}}$ for each variant. The dashed red line denotes wild-type activity. Asterisks indicate variants significant after Benjamini–Hochberg FDR correction ($q < 0.05$). Variants 7 (S175E;D197F;K214Q;G232Y) and 9 (E5Y;A87Y;D117A;M153F;R168L), both proposed by the `RAW SEQ / MLP` model, are the only constructs to exceed wild-type fluorescence, achieving 3- to 4-fold higher activity. Remaining variants performed around or below wild-type levels. Variants falling below 10% of the wild-type mKate median expression are not shown.

The majority of generated variants exhibited little detectable fluorescence in wet-lab validation. This outcome is consistent with the extreme sensitivity of fluorescent proteins to destabilizing mutations: even small sequence changes can disrupt chromophore maturation or overall structural stability, leading to complete loss of function. While the oracles were trained to predict fluorescence from sequence, the results suggest that they did not fully capture the underlying biophysical constraints required for maintaining foldability under combinatorial mutation. This challenge likely becomes more severe as mutational distance from wild type increases, where non-local, epistatic effects dominate.

Variants proposed using the RAW SEQ / PF MLP oracle produced the strongest fluorescent hits, despite this oracle not being the top performer on held-out predictive benchmarks. We caution against overinterpreting this observation: with only seven variants generated per oracle, the per-architecture wet-lab sample is too small to support firm claims about architectural ranking, and the result should be read as suggestive rather than conclusive.

That said, a divergence between oracle predictive accuracy and the rate of experimentally validated hits emerging from RL-guided design is consistent with known properties of policy optimization against learned reward models. Two factors are particularly relevant here. First, our oracle benchmarks evaluate in-distribution prediction, largely on single mutants, whereas the RL policy generates combinatorial mutants that lie outside the support of the DMS training data; strong in-distribution accuracy does not guarantee faithful extrapolation to the regions of sequence space the policy actually explores. Second, when an oracle is used as a reward signal, the agent is incentivized to seek out inputs that score highly under the learned reward, which need not coincide with inputs that satisfy the underlying biophysical constraints. Higher-capacity architectures plausibly offer a larger surface for this kind of reward exploitation, a manifestation of Goodhart's law for learned rewards. Under this view, the relative robustness of the MLP-derived proposals reflects a property of the optimization procedure rather than evidence that simpler oracles capture the fitness landscape more faithfully.

Disentangling these effects, for example through reward-model ensembling, uncertainty-aware policy regularization, or constraining proposals to the support of the training distribution, is an important direction for future work and may allow more expressive oracles to translate their predictive gains into experimental gains.

6. Discussion and Conclusion

We introduced AURORA, a protein engineering framework that decouples representation learning from mutation proposal by combining MSA-based oracles with reinforcement learning. Our results show that this separation enables more effective modeling of epistasis, broader exploration of sequence space, and competitive discovery of high-fitness variants with limited experimental data.

Across oracle ablations, MSA Pairformer representations consistently outperformed single-sequence ESM embeddings, supporting the hypothesis that explicitly observing co-evolutionary context provides stronger inductive biases for variant effect prediction. This advantage persisted across oracle architectures and translated into measurable differences in downstream RL behavior. In particular, policies trained with MSA Pairformer-based oracles explored the action space more broadly than those trained with ESM-based oracles, an effect most pronounced for simpler MLP policies though attenuated when using more expressive BiMamba backbones. These results indicate that oracle choice shapes not only predictive accuracy but also the effective fitness landscape encountered during policy optimization.

Decoupling representation from search addresses a key architectural tension in protein engineering: evolutionary models describe typical sequences, whereas experimental objectives target rare, high-performing variants. By fixing the oracle and training a separate policy, AURORA avoids forcing a single model to serve both roles. This design enables direct multi-site mutation proposals via a joint action space, in contrast to iterative single-site or autoregressive approaches that struggle with epistasis. In avGFP, this formulation led to the discovery of 42 novel variants with predicted fitness well above the DMS distribution, with mutation patterns that diverge from the training data.

Our study has limitations. Experimental results are currently restricted to avGFP, a well-characterized system with dense DMS coverage. Additionally, Pairformer performance depends on MSA quality, and PPO training was intentionally truncated to preserve diversity. Future work will explore hybrid representations, improved exploration objectives, and tighter integration with active learning.

Overall, AURORA demonstrates that architectural specialization—MSA-based models for representation and RL for search—offers a principled and effective alternative to monolithic protein engineering pipelines. We expect this decoupled design to generalize across proteins and objectives, enabling more efficient navigation of complex fitness landscapes.

Software and Data

Source code and data analysis scripts used in this study are available at https://github.com/katie-sp/aurora.

Acknowledgements

We thank Lyra Labs for compute and MIT's Huang-Hobbs BioMaker Space for providing bench space and materials for our wet lab experimentation. We thank our anonymous reviewers for their time and constructive feedback, which greatly helped us improve the clarity and quality of our manuscript.

Impact Statement

This work aims to advance machine learning for protein engineering by improving how computational methods guide experimental design. The primary impact is reducing the experimental cost of developing proteins for therapeutic, agricultural, and industrial applications. As with all protein engineering research, our methods could theoretically be applied to both beneficial and harmful ends. However, the approach requires significant wet-lab expertise and resources, which naturally limits misuse. We encourage responsible use and transparent communication of these methods within the scientific community.

References

Akiyama, Y., Zhang, Z., Mirdita, M., Steinegger, M., and Ovchinnikov, S. Scaling down protein language modeling with msa pairformer. *bioRxiv*, 2025. doi: 10.1101/2025.08.02.668173. URL https://www.biorxiv.org/content/early/2025/08/03/2025.08.02.668173.

Benjamins, C., Surana, S., Bent, O., Lindauer, M., and Duckworth, P. Bayesian optimization for protein sequence design: Back to simplicity with gaussian processes. In *AI for Accelerated Materials Design - NeurIPS 2024*, 2024. URL https://openreview.net/forum?id=zori4pHmFP.

Cao, H., Zhang, H., Xu, J., Zhang, Z., Shen, L., Sun, M., Liu, G., Xu, J., Li, W.-J., Ni, J., de la Fuente-Nunez, C., Fu, T., Choi, Y., Heng, P.-A., and Wu, F. From supervision to exploration: What does protein language model learn during reinforcement learning?, 2025. URL https://arxiv.org/abs/2510.01571.

Cobb, R. E., Chao, R., and Zhao, H. Directed evolution: Past, present, and future. *AIChE Journal*, 59(5): 1432–1440, 2013. doi: https://doi.org/10.1002/aic.13995. URL https://aiche.onlinelibrary.wiley.com/doi/abs/10.1002/aic.13995.

Dao, T. and Gu, A. Transformers are ssms: Generalized models and efficient algorithms through structured state space duality. *arXiv preprint arXiv:2405.21060*, 2024.

Epstein, C. J., Goldberger, R. F., and Anfinsen, C. B. The genetic control of tertiary protein structure: studies with model systems. In *Cold Spring Harbor Symposia on Quantitative Biology*, volume 28, pp. 439–449. Cold Spring Harbor Laboratory Press, 1963.

Hie, B. L., Xu, D., Shanker, V. R., Bruun, T. U., Weidenbacher, P. A., Tang, S., and Kim, P. S. Efficient evolution of human antibodies from general protein language models. *Nature*, 616:281–288, 2023. doi: 10.1038/s41587-023-01763-2.

Hou, C., Liu, D., Zafar, A., and Shen, Y. Understanding language model scaling on protein fitness prediction. *bioRxiv*, 2025. doi: 10.1101/2025.04.25.650688. URL https://www.biorxiv.org/content/early/2025/07/23/2025.04.25.650688.

Jiang, K., Yan, Z., Bernardo, M. D., Sgrizzi, S. R., Villiger, L., Kayabolen, A., Kim, B. J., Carscadden, J. K., Hiraizumi, M., Nishimasu, H., Gootenberg, J. S., and Abudayyeh, O. O. Rapid in silico directed evolution by a protein language model with evolvepro. *Science*, 387(6732):eadr6006, 2025. doi: 10.1126/science.adr6006. URL https://www.science.org/doi/abs/10.1126/science.adr6006.

Johnston, K. E., Almhjell, P. J., Watkins-Dulaney, E. J., Liu, G., Porter, N. J., Yang, J., and Arnold, F. H. A combinatorially complete epistatic fitness landscape in an enzyme active site. *Proceedings of the National Academy of Sciences*, 121(32):e2400439121, 2024. doi: 10.1073/pnas.2400439121. URL https://www.pnas.org/doi/abs/10.1073/pnas.2400439121.

Lin, Z., Akin, H., Rao, R., Hie, B., Zhu, Z., Lu, W., Smetanin, N., Verkuil, R., Kabeli, O., Shmueli, Y., dos Santos Costa, A., Fazel-Zarandi, M., Sercu, T., Candido, S., and Rives, A. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023. doi: 10.1126/science.ade2574. URL https://www.science.org/doi/abs/10.1126/science.ade2574.

Madani, A., Krause, B., Greene, E. R., Subramanian, S., Mohr, B. P., Holton, J. M., Olmos Jr, J. L., Xiong, C., Sun, Z. Z., Socher, R., et al. Large language models generate functional protein sequences across diverse families. *Nature biotechnology*, 41(8):1099–1106, 2023.

Notin, P., Dias, M., Frazer, J., Marchena-Hurtado, J., Gomez, A., Marks, D. S., and Gal, Y. Tranception: Protein fitness prediction with autoregressive transformers and inference-time retrieval. In Chaudhuri, K., Jegelka, S., Song, L., Szepesvari, C., Niu, G., and Sabato, S. (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 16990–17017. PMLR, 2022.

Notin, P., Kollasch, A. W., Ritter, D., van Niekerk, L., Paul, S., Spinner, H., Rollins, N., Shaw, A., Weitzman, R., Frazer, J., Dias, M., Franceschi, D., Orenbuch, R., Gal, Y., and Marks, D. S. ProteinGym: Large-scale benchmarks for protein fitness prediction and design. In *Advances in Neural Information Processing Systems*, volume 36, pp. 64331–64379. Curran Associates, Inc., 2023.

Rao, R. M., Liu, J., Verkuil, R., Meier, J., Canny, J., Abbeel, P., Sercu, T., and Rives, A. Msa transformer. In Meila, M. and Zhang, T. (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8844–8856. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/rao21a.html.

Rives, A., Meier, J., Sercu, T., Goyal, S., Lin, Z., Liu, J., Guo, D., Ott, M., Zitnick, C. L., Ma, J., and Fergus, R. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proceedings of the National Academy of Sciences*, 118(15):e2016239118, 2021. doi: 10.1073/pnas.2016239118.

Sarkisyan, K. S., Bolotin, D. A., Meer, M. V., Usmanova, D. R., Mishin, A. S., Sharonov, G. V., Ivankov, D. N., Bozhanova, N. G., Baranov, M. S., Soylemez, O., Bogatyreva, N. S., Vlasov, P. K., Egorov, E. S., Logacheva, M. D., Kondrashov, A. S., Chudakov, D. M., Putintseva, E. V., Mamedov, I. Z., Tawfik, D. S., Lukyanov, K. A., and Kondrashov, F. A. Local fitness landscape of the green fluorescent protein. *Nature*, 533(7603):397–401, 2016. doi: 10.1038/nature17995. URL https://doi.org/10.1038/nature17995.

Soldát, M. and Kléma, J. Directed evolution of proteins via bayesian optimization in embedding space. In *2024 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pp. 91–98. IEEE, December 2024. doi: 10.1109/bibm62325.2024.10822356. URL http://dx.doi.org/10.1109/BIBM62325.2024.10822356.

Starr, T. N. and Thornton, J. W. Epistasis in protein evolution. *Protein Science*, 25(7):1204–1218, 2016. doi: 10.1002/pro.2897. URL https://pmc.ncbi.nlm.nih.gov/articles/PMC4918427/.

Stocco, F., Artigues-Lleixa, M., Hunklinger, A., Widatalla, T., Guell, M., and Ferruz, N. Guiding generative protein language models with reinforcement learning, 2025. URL https://arxiv.org/abs/2412.12979.

Sun, H., He, L., Deng, P., Liu, G., Zhao, Z., Jiang, Y., Cao, C., Ju, F., Wu, L., Liu, H., Qin, T., and Liu, T.-Y. Accelerating protein engineering with fitness landscape modelling and reinforcement learning. *Nature Machine Intelligence*, 7(9):1446–1460, 2025. doi: 10.1038/s42256-025-01103-w. URL https://www.nature.com/articles/s42256-025-01103-w.

Tokuriki, N. and Tawfik, D. S. Stability effects of mutations and protein evolvability. *Current Opinion in Structural Biology*, 19(5):596–604, 2009. ISSN 0959-440X. doi: https://doi.org/10.1016/j.sbi.2009.08.003. URL https://www.sciencedirect.com/science/article/pii/S0959440X09001249. Carbohydradtes and glycoconjugates / Biophysical methods.

Truong Jr, T. and Bepler, T. Poet: A generative model of protein families as sequences-of-sequences. *Advances in Neural Information Processing Systems*, 36:77379–77415, 2023.

Widatalla, T., Rafailov, R., and Hie, B. Aligning protein generative models with experimental fitness via direct preference optimization. *arXiv preprint arXiv:2402.04897*, 2024.

Wittmann, B. J., Yue, Y., and Arnold, F. H. Informed training set design enables efficient machine learning-assisted directed protein evolution. *Cell Systems*, 12(11):1026–1045.e7, 2021. ISSN 2405-4712. doi: https://doi.org/10.1016/j.cels.2021.07.008. URL https://www.sciencedirect.com/science/article/pii/S2405471221002866.

Yang, K. K., Wu, Z., and Arnold, F. H. Machine-learning-guided directed evolution for protein engineering. *Nature Methods*, 16(8):687–694, 2019. doi: 10.1038/s41592-019-0496-6. URL https://doi.org.

Zhang, Z., Wayment-Steele, H. K., Brixi, G., Wang, H., Kern, D., and Ovchinnikov, S. Protein language models learn evolutionary statistics of interacting sequence motifs. *Proceedings of the National Academy of Sciences*, 121(45):e2406285121, 2024.

A. Appendix

A.1. Pairformer-based PPO exploration

MSA Pairformer-based training leads to broader action visitation compared to ESM-based training.

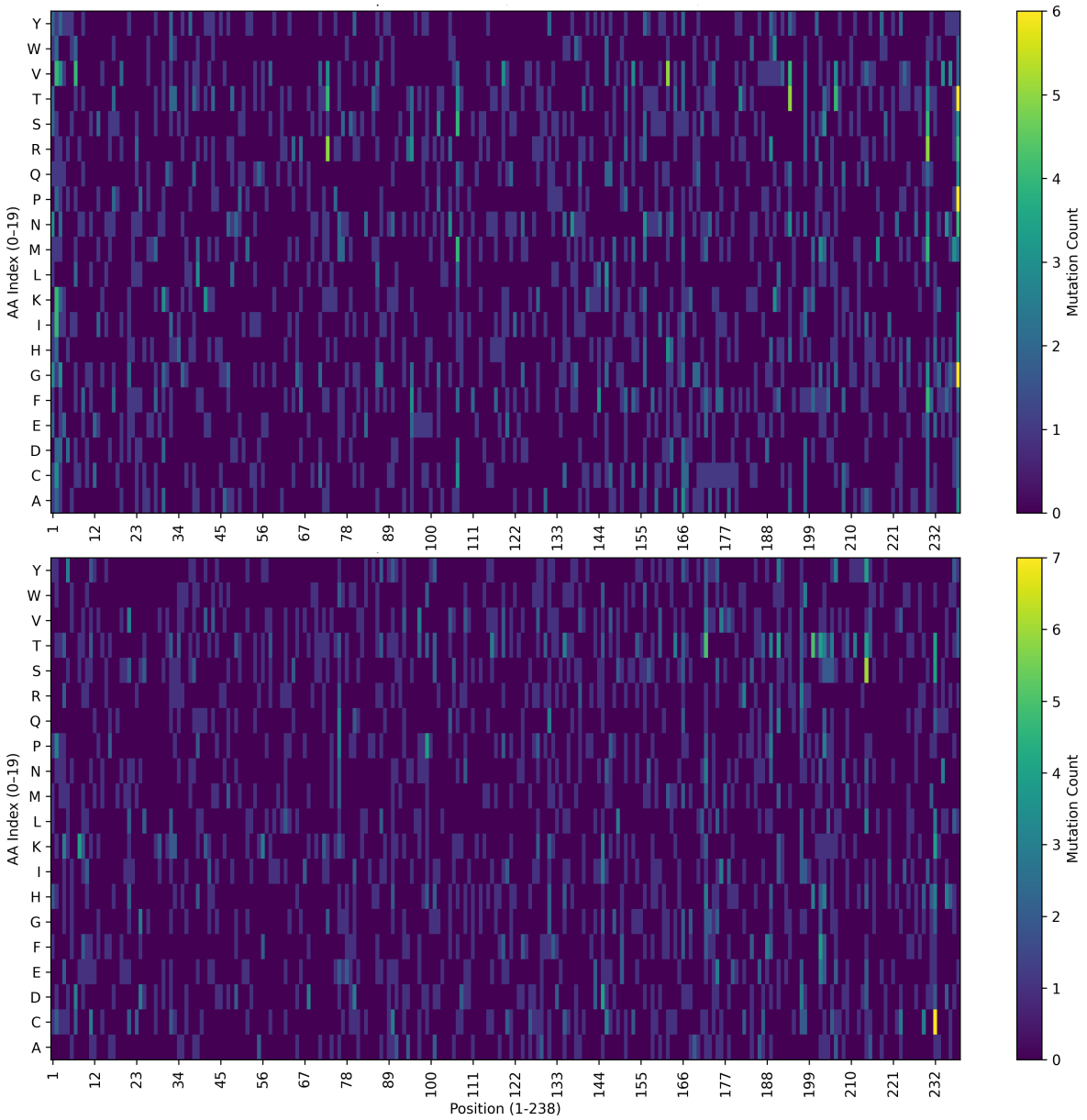

Figure 6. **Heatmaps of action space visitation frequency during PPO training.** The top panel corresponds to training with a `Pairformer / BiMamba` oracle, while the bottom panel corresponds to an `ESM / BiMamba` oracle. By visual inspection, the `Pairformer / BiMamba`–guided policy exhibits slightly broader exploration across the action space. Note that the color scale differs between panels, with the `Pairformer / BiMamba` heatmap spanning a smaller range, reflecting a more distributed visitation pattern.

A.2. Novelty of mutants proposed by each RL model

RL models guided by various oracles tend to propose different mutational landscapes.

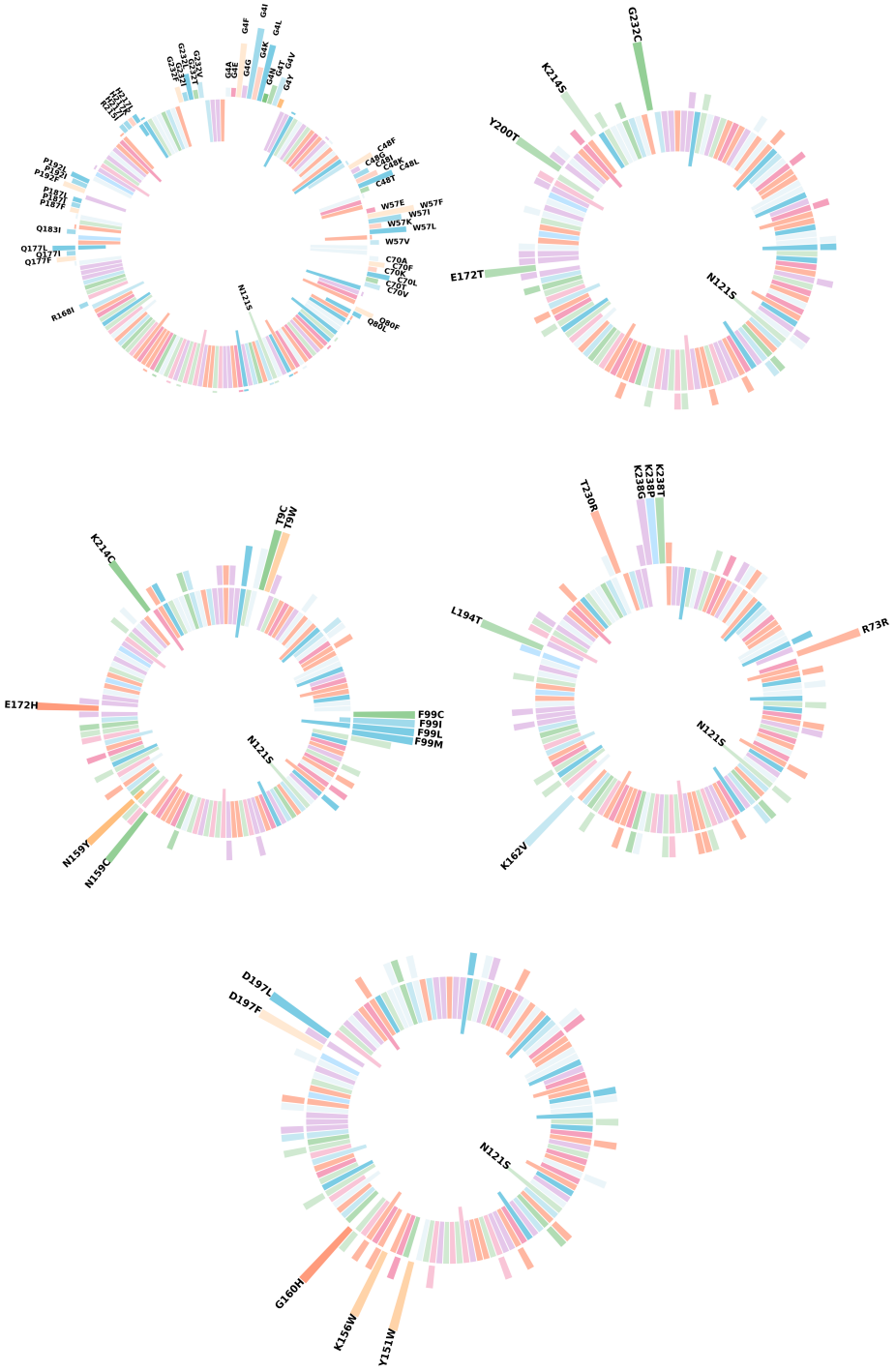

Figure 7. **RL-proposed mutations are novel compared to DMS dataset.** Comparison of mutation frequencies between the DMS (inner ring) and mutations proposed by each RL model (outer ring); from left to right and top to bottom, RL models based on the `ESM / MLP` oracle, `ESM / BiMamba` oracle, `PF / MLP` oracle, `PF / BiMamba` oracle, and `RAW / BiMamba` oracle. Each bar represents a single-point mutation. Bar height indicates mutation frequency as a percentage of total mutations in each dataset, with colors denoting amino acid identity. Mutations appearing at $\geq$0.3% frequency in at least one dataset are shown.

A.3. Categorical Jacobian Analysis

To interpret how each model captures position-wise dependencies in protein sequences, we compute a *categorical Jacobian* that measures how perturbing the amino acid identity at one position affects the model's predicted amino acid logits at all other positions.

A.3.1. Definition

Let $\mathbf{x} = (x_1, \ldots, x_L)$ denote an input protein sequence of length L, where each $x_i \in \{1, \ldots, 20\}$ indexes an amino acid. Each model produces per-position logits $f(\mathbf{x}) \in \mathbb{R}^{L \times 20}$ via a learned output projection from hidden states.

The categorical Jacobian $\mathbf{J} \in \mathbb{R}^{L \times 20 \times L \times 20}$ is defined as:

$$J_{i,a,j,b} = f_{j,b}(\mathbf{x}^{(i \to a)}) - f_{j,b}(\mathbf{x}) \tag{8}$$

where $\mathbf{x}^{(i \to a)}$ denotes the sequence with position i substituted by amino acid a, and $f_{j,b}(\cdot)$ is the logit for amino acid b at position j. This measures how substituting amino acid a at position i changes the predicted logit for amino acid b at position j.

A.3.2. Computation

For each position $i \in \{1, \ldots, L\}$, we generate 20 perturbed sequences (one per amino acid substitution), compute forward passes through the model, and subtract the reference logits. This yields $L \times 20$ forward passes per sequence. A faster approximation uses only mask-token perturbations ($n = 1$ instead of $n = 20$), reducing computation by a factor of 20.

A.3.3. Derived Quantities

Coevolution Map. We collapse the 4D Jacobian to a 2D position-position coupling matrix by computing the Frobenius norm over amino acid dimensions:

$$C_{ij} = \|\mathbf{J}_{i,:,j,:}\|_F = \sqrt{\sum_{a,b} J_{i,a,j,b}^2} \tag{9}$$

after centering along each dimension. The matrix is symmetrized as $C \leftarrow (C + C^\top)/2$ with diagonal set to zero. We optionally apply Average Product Correction (APC) to remove phylogenetic background:

$$C_{ij}^{\text{APC}} = C_{ij} - \frac{C_{i\cdot}\, C_{\cdot j}}{C_{\cdot\cdot}} \tag{10}$$

where $C_{i\cdot}$, $C_{\cdot j}$, and $C_{\cdot\cdot}$ denote row, column, and total sums respectively. Higher values of C_{ij} indicate stronger functional coupling between positions i and j.

Conservation Score. Per-position sensitivity is computed as the total magnitude of Jacobian entries originating from each position:

$$S_i = \sqrt{\sum_{a,j,b} J_{i,a,j,b}^2} \tag{11}$$

Positions with high S_i values have large effects on the model's predictions when mutated, suggesting functional importance.

A.3.4. Interpretation

The categorical Jacobian reveals how models encode position-wise dependencies:

- **Bidirectional models** can capture symmetric pairwise interactions, as each position has access to full sequence context.
- **Autoregressive/causal models** produce asymmetric Jacobians where perturbations at position i only affect positions $j > i$, reflecting the causal structure.
- **Coevolution maps** highlight residue pairs that the model treats as functionally coupled, which may correspond to structural contacts or epistatic interactions.

- **Conservation profiles** identify positions where the model is most sensitive to mutations, potentially corresponding to catalytic sites or structurally critical residues.