MUTAGEN: IMPLICITLY GUIDED PROTEIN EVOLUTION FROM RANKED FEEDBACK VIA PAIR-BASED DISCRETE FLOW MATCHING

Anonymous authors
Paper under double-blind review

ABSTRACT

Machine learning-directed evolution (MLDE) (Yang et al., 2019) aims at democratizing protein engineering, enabling optimization of any protein with any assay at accessible cost by drastically reducing the screening of thousands of protein sequences. In this work, we introduce a novel discrete flow-matching (DFM) method, MUTAGEN, trained to iteratively mutate protein sequences towards high-fitness regions of the protein fitness landscape, without relying on noisy in-silico fitness predictions. Training minimizes a token-level cross-entropy flow-matching loss to learn a vector field of improvement from ranked sequence pairs alone. Across realistic screening budgets, MUTAGEN enables multi-mutational protein optimization with minimal data (as low as 20 sequences per round of evolution) while bypassing the need for an explicit fitness predictor (Yang et al., 2025). We validate our approach on standard in silico benchmarks (GFP and AAV) (Sarkisyan et al., 2016; Bryant et al., 2021) and experimentally in a four-round campaign on NanoLuc, achieving an >80-fold increase in luminescence over the wild-type.

1 INTRODUCTION

Directed evolution is a cornerstone of modern protein engineering (Arnold, 1998; 2018; Packer & Liu, 2015). Significant effort has been devoted to machine learning methods that reduce the experimental burden by proposing variants more efficiently than classical heuristics (Jiang et al., 2024; Hie et al., 2024). In practice, however, many current approaches struggle to reliably identify high-fitness multi-mutants in the data-limited regimes that dominate real campaigns (Yang et al., 2019; 2025).

In this work, we introduce MUTAGEN, a discrete flow-matching (DFM) model (Campbell et al., 2024) that learns to transform low-fitness protein sequences into high-fitness variants using only *ranked pairs*. In realistic directed evolution campaigns, practitioners can reliably determine which variants outperform others, but absolute fitness measurements are often corrupted by plate effects, batch variability, and assay noise (Packer & Liu, 2015; Yang et al., 2025). Existing machine learning-directed evolution (MLDE) methods typically train surrogate models on these noisy scalar measurements and then optimize against the learned landscape (Yang et al., 2019; Jiang et al., 2024; Kirjner et al., 2024)–an approach that degrades rapidly when fitness regression becomes unreliable. MUTAGEN sidesteps this failure mode entirely: given ranked pairs of protein sequences, it learns an implicit *vector field of improvement* directly from pairwise comparisons, requiring no explicit fitness values.

2 METHOD

2.1 MUTAGEN: IMPLICITLY GUIDED PAIR-BASED DISCRETE FLOW MATCHING

A protein sequence is represented as a vector of categorical tokens $x \in \{1, \dots, K\}^L$. We model a "flow" over discrete sequences as a family of categorical distributions evolving over time under a continuous-time Markov chain (CTMC), following discrete flow matching and related discrete-state generative formulations (Campbell et al., 2024).

MUTAGEN learns a pair-conditioned categorical flow. Given a low-fitness sequence x_L and a higher-fitness sequence x_H, we train a CTMC generator that moves probability mass from x_L to x_H.

2.1.1 PAIRWISE DATA SAMPLING

From the set of sequences $x_i \in G$ we have available in each round r we form the following:

$$P = \Big\{(x_{\mathrm{L}}, x_{\mathrm{H}}) \,:\, f(x_{\mathrm{L}}) < f(x_{\mathrm{H}}),\ d_H(x_{\mathrm{L}}, x_{\mathrm{H}}) \le D_{\max}\Big\}.$$

Given a pair we sample $t \sim \mathrm{U}(0,1)$ and create $x_t^{(j)} = \big\{x_{\mathrm{H}}^{(j)} \text{ w.p. } t,\ x_{\mathrm{L}}^{(j)} \text{ otherwise}\big\}$. The conditional target is the mixture $q_t = \mathrm{Cat}\big(t\,\delta_{x_{\mathrm{H}}} + (1-t)\,\delta_{x_{\mathrm{L}}}\big)$. Its optimal generator satisfies $R_t^*(x_{\mathrm{L}} \to x_{\mathrm{H}}) = (1-t)^{-1}$ (Campbell et al., 2024).

2.1.2 NEURAL GENERATOR

We parameterize the rate field with a time-aware sequence encoder. Specifically, protein tokens are first embedded by an ESM-2 8M backbone model (Lin et al., 2023), yielding a 320-dimensional representation $h(x_t)$. A sinusoidal embedding of the scalar time t is concatenated with $h(x_t)$ and passed through a two–layer MLP that outputs logits $g_\theta(x_t, t) \in \mathbb{R}^{L\times|\Sigma|}$. The induced categorical proposal is

$$\pi_\theta^{(j)}(\cdot \mid x_t, t) = \mathrm{softmax}\,\big(g_\theta^{(j,\cdot)}\big),$$

which serves as the neural approximation to the optimal generator in the discrete flow-matching objective.

2.1.3 FLOW-MATCHING LOSS

For small time steps Δt, the discrete flow-matching objective reduces to a cross-entropy token loss (Campbell et al., 2024). We minimize the expected difference between the model's vector field and the conditional vector field generating the path between samples:

$$\mathcal{L}_{\mathrm{FM}} = \mathbb{E}_{(x_{\mathrm{L}},x_{\mathrm{H}}),\,t}\Big[\sum_{j=1}^{L} \mathrm{CE}\big(\pi_\theta^{(j)}(\cdot \mid x_t, t),\, x_{\mathrm{H}}^{(j)}\big)\Big]. \tag{1}$$

This trains the model to predict the high-ranked tokens from interpolated sequences. Conceptually, the model learns "how to rewrite low-ranked sequences into higher-ranked ones" using only ordered pairs.

2.1.4 LOCATION-AWARE SPARSITY

Mutating large portions of proteins often leads to non-functional mutants. To resolve this issue we modified our algorithm and included additional loss terms:

1. We add a term $\lambda_1\|g_\theta\|_1$ to the loss that discourages dense long-range jumps in between interpolated states, leading to the objective
$$\mathcal{L} = \mathcal{L}_{\mathrm{FM}} + \lambda_1\|g_\theta\|_1.$$
2. We sample a fixed k mutation locations every round proportional to the softmax of ESM-2 attention scores computed on the wildtype sequence. This mirrors the saliency map based approach of Gruver et al. (2023). This model is not finetuned on fitness values, but rather the native attention scores of the ESM-2 model provide a powerful prior of mutation locations.

For a controlled study applying the same saliency mask to all baselines, see Appendix §H (Fig. 7). For hyperparameter settings of MUTAGEN and baselines, see Appendix §I–§K.

2.1.5 SAMPLING

We integrate the learned CTMC with $N = 10$ uniform steps. At each $t_i = i/N$ we

1. Compute logits $g_\theta(x_{t_i}, t_i)$,

2. Obtain rates via the discrete Fokker–Planck rule of Campbell et al. (2024),
3. Accept at most one substitution per site with probability $\min\{1, R_\theta \Delta t\}$,
4. Freeze positions outside the saliency mask.

For pseudocode, see Algorithm 1 (Appendix §C).

2.2 BASELINES

We compare methods that propose sequences under the same per-round screening budget B and use the same evaluation mechanism. Specifically, we study four acquisition strategies: RANDOM (uniform proposal), EVOLVEPRO Jiang et al. (2024), MCMC, and MUTAGEN (our flow-matching sampler, Alg. 1). The surrogate-based methods (EvolvePro, MCMC) are trained on scalar fitness values, whereas MUTAGEN uses only ranked comparisons derived from those evaluations to select mutations–a strictly weaker form of supervision. Full experimental workflow details are in Appendix §B.

3 RESULTS

We evaluate MUTAGEN in three settings: *in silico* optimization on GFP (Sarkisyan et al., 2016) and AAV (Bryant et al., 2021) protein fitness landscapes, and a four-round wet-lab optimization of NanoLuc (Hall et al., 2012), a bright bioluminescent enzyme engineered from deep-sea shrimp. NanoLuc provides a quantifiable, easily measurable phenotype with a simple and rapid assay, making it an ideal target for directed evolution campaigns. This wet-lab validation demonstrates that MUTAGEN's learned model generalizes to real experimental settings beyond *in silico* benchmarks. *In silico*, we evaluate candidates using a frozen neural oracle f_{oracle}; in the wet lab, we evaluate candidates using luminescence measurements (see G). For full oracle and label construction details, see Appendix §B.

3.1 BENCHMARKS

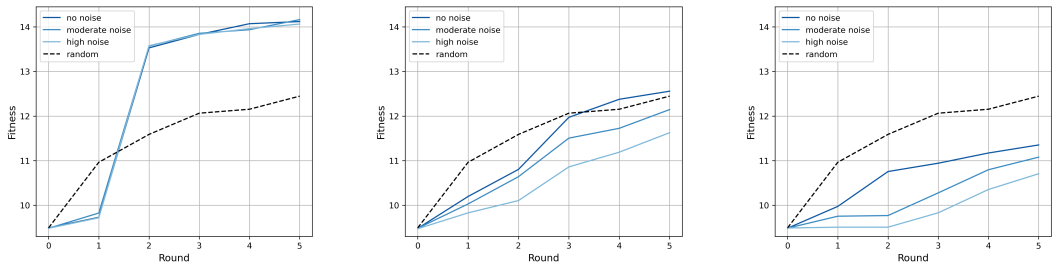
Table 1: Maximum fitness achieved as measured by corresponding oracle ($N = 20$).

METHOD	GFP	AAV
RANDOM	3.79 ± 0.04	12.45 ± 1.36
EVOLVEPRO	$\mathbf{3.83 \pm 0.06}$	12.56 ± 1.91
MCMC	3.76 ± 0.05	11.35 ± 1.89
MUTAGEN	$\mathbf{3.80 \pm 0.03}$	$\mathbf{14.12 \pm 0.64}$

Table 1 summarizes performance across 20 independent runs (each starting from a different random seed) at a strict per-round budget of $N = 20$. For the corresponding fitness-vs-round curves, see Figure 3 in Appendix §D.

We find that on the AAV protein optimization task, MUTAGEN strongly outperforms all other methods, achieving a maximum fitness of (14.12 ± 0.64), compared to EvolvePro (12.56 ± 1.91), random sampling (12.45 ± 1.36), and MCMC (11.35 ± 1.89). Critically, both EvolvePro and MCMC fail at surpassing the simple baseline of random mutagenesis on AAV. MUTAGEN is the only method that is able to achieve substantial fitness gains.

On the GFP protein optimization task, all methods exhibit only modest improvements across rounds. MUTAGEN achieves a mean maximum fitness of (3.80 ± 0.03), slightly exceeding random sampling (3.79 ± 0.04). MCMC performs worse than the random baseline (3.76 ± 0.05), whereas EvolvePro attains the highest mean maximum (3.83 ± 0.06). Importantly, on GFP the minimal difference between EvolvePro and MUTAGEN indicates that preference-only feedback is sufficient to reach near-parity with state of the art scalar-supervised baselines, while leading to even stronger performance on the more challenging AAV landscape.



(a) MutaGen, AAV (b) EvolvePro, AAV (c) MCMC, AAV

Figure 1: AAV robustness to noisy fitness measurements. Closed-loop optimization trajectories under additive noise in the feedback signal (no / moderate / high), with the random baseline shown as a dashed line. Columns correspond to MUTAGEN (left), EvolvePro (middle), and MCMC (right). Curves show the mean cumulative best fitness over 20 runs.

Considering the results across both oracles, MUTAGEN is the only method that is able to consistently outperform the baseline of random mutagenesis across multiple proteins. For additional budget allocation analysis (round-size tradeoff curves and full numerical sweeps), see Appendix §E.

3.2 ROBUSTNESS TO NOISE

A central motivation for rank-based optimization is that real wet-lab measurements are often noisy: plate effects, batch-to-batch variability, and assay stochasticity can distort absolute fitness values even when relative ordering is partially preserved. To evaluate robustness under such conditions, we corrupt the per-round feedback signal with additive measurement noise before constructing rankings (for MUTAGEN) or fitting score-based surrogates (for EvolvePro and MCMC). Concretely, for each evaluated sequence x with oracle score $f_{\text{oracle}}(x)$, we form a noisy observation $\tilde{f}(x) = f_{\text{oracle}}(x) + \epsilon$ with $\epsilon \sim \mathcal{N}(0, \sigma^2)$, and vary σ across *no*, *moderate*, and *high* noise regimes. The exact σ settings are given in Appendix §F. We then run the full closed-loop optimization campaign under the noisy feedback, while reporting best-achieved fitness on the original noiseless oracles.

Figure 1 shows that MUTAGEN is consistently robust to substantial noise on AAV. On AAV, MUTAGEN continues to rapidly reach high fitness scores and is largely unaffected by the increase in noise, whereas score-based baselines degrade markedly under moderate and high noise, indicating that surrogate-guided acquisition becomes unreliable when the regression target is corrupted at low sample sizes. For the corresponding GFP robustness trajectories, see Appendix Fig. 5.

3.3 WET-LAB NANOLUC OPTIMIZATION

In order to compare MUTAGEN, EvolvePro, MCMC and the random baseline in an experimental setting, we performed a four-round optimization campaign on NanoLuc (20 mutants per round, details in G) which produces intense bioluminescence upon oxidation of its substrate furimazine. Figure 2a shows the luminescence fold-change improvements for all methods over the different rounds.

Across experimental rounds, MUTAGEN is the first method to identify a mutant exceeding an 80-fold increase in luminescence relative to wild-type NanoLuc, achieving this in round 3 of evolution and producing the highest-luminescence variant observed across all methods. In addition, MUTAGEN consistently yields the largest fraction of high-fitness mutants in rounds 3 and 4. By round 4, both the random strategy and EvolvePro approach the peak performance achieved by MUTAGEN, while MUTAGEN shifts toward exploration of nearby but less luminescent variants.

We repeated measurements of high-fitness variants across rounds of evolution. These repeats reveal substantial plate-to-plate variability: replicate measurements differ by a median factor of 1.6, which limits the reliability of using a single highest-fitness measurement as the primary evaluation metric, as is commonly done in in silico benchmarks.

To more robustly assess model performance under this high-noise regime, we therefore evaluate the fraction of high-fitness mutants produced by each method in each round (Fig. 2b). Under this metric,

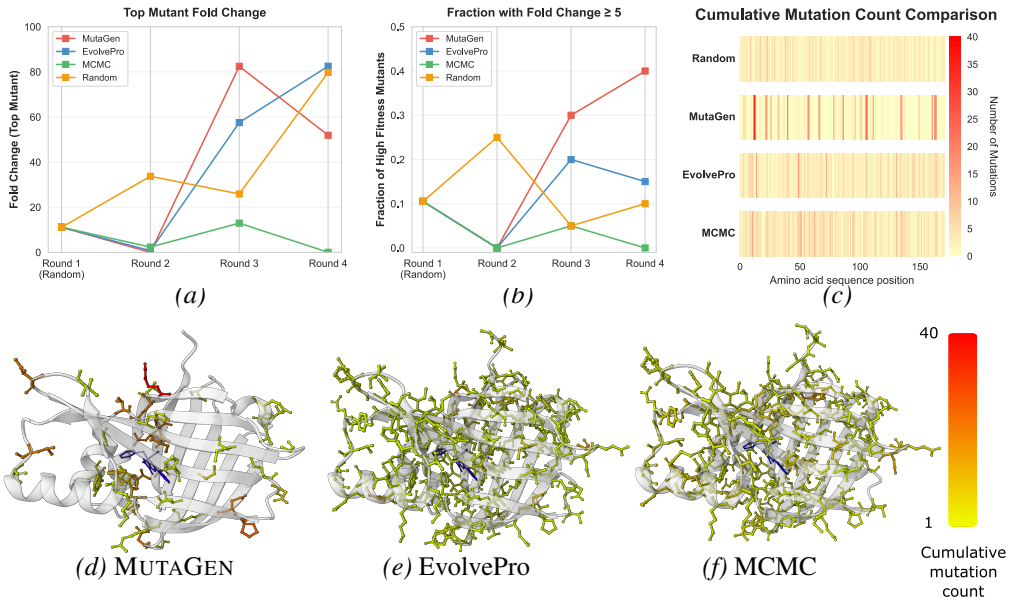


Figure 2: NanoLuc wet-lab results and mutation patterns. (a) Fold-change of top-mutant luminescence relative to wild-type in each round. (b) Fraction of assayed mutants with fold-change ≥ 5 in each round. (c) Cumulative mutation counts per residue position across the campaign for each method. (d–f) Mutated residues mapped onto the Boltz2 predicted NanoLuc structure for MUTAGEN, EvolvePro, and MCMC, with the furimazine ligand shown in blue Passaro et al. (2025); Wohlwend et al. (2025).

MUTAGEN substantially outperforms all other methods in both rounds 3 and 4. These results suggest that, in realistic protein optimization campaigns with noisy experimental feedback, MUTAGEN is the most reliable approach for consistently generating high-fitness variants. For mutation localization and active-site proximity analyses, see Appendix §G.5.

4 CONCLUSION

We introduced MUTAGEN, a pair-conditioned discrete flow-matching method for machine learning-directed protein evolution that learns an implicit *vector field of improvement* from ranked sequence pairs without training an explicit fitness predictor, modeling sequence edits as a CTMC over categorical tokens and training with a token-level flow-matching cross-entropy objective on interpolants. Across realistic campaigns, MUTAGEN consistently outperforms random mutagenesis on multiple proteins, with especially strong gains on AAV where scalar-supervised methods often fail to beat the random baseline, and it remains robust under substantial measurement noise. A four-round wet-lab NanoLuc campaign corroborates these advantages, yielding an $>$ 80-fold improvement over wild-type and a higher fraction of strongly improved variants despite plate-to-plate variability, highlighting pair-based discrete flow matching as a practical alternative to surrogate-guided MLDE in data-limited, noisy regimes. For additional analyses (round-size tradeoffs, saliency ablations, and hyperparameters), see Appendix §E–§K.

REFERENCES

Frances H. Arnold. Design by directed evolution. *Accounts of Chemical Research*, 31(3):125–131, 1998. doi: 10.1021/ar960017f. URL https://doi.org/10.1021/ar960017f.

Frances H. Arnold. Directed evolution: Bringing new chemistry to life. *Angewandte Chemie International Edition*, 57(16):4143–4148, 2018. doi: 10.1002/anie.201708408. URL https://doi.org/10.1002/anie.201708408.

Daniel H. Bryant et al. Deep diversification of an AAV capsid protein by machine learning. *Nature Biotechnology*, 2021. URL https://www.nature.com/articles/s41587-020-00793-4. Dataset paper used in the AAV benchmark.

Andrew Campbell, Jason Yim, Regina Barzilay, Tom Rainforth, and Tommi Jaakkola. Generative flows on discrete state-spaces: Enabling multimodal flows with applications to protein co-design. *arXiv preprint arXiv:2402.04997*, 2024. URL https://arxiv.org/abs/2402.04997.

James Dunbar and Charlotte M. Deane. Anarci: antigen receptor numbering and receptor classification. *Bioinformatics*, 32(2):298–300, 2016. doi: 10.1093/bioinformatics/btv552. URL https://doi.org/10.1093/bioinformatics/btv552.

Nate Gruver, Samuel Don Stanton, Nathan C. Frey, Tim G. J. Rudner, Isidro Hotzel, Julien Lafrance-Vanasse, Arvind Rajpal, Kyunghyun Cho, and Andrew Gordon Wilson. Protein design with guided discrete diffusion. In *Advances in Neural Information Processing Systems*, 2023.

Michael P. Hall et al. Engineered luciferase reporter from a deep sea shrimp utilizing a novel imidazopyrazinone substrate. *ACS Chemical Biology*, 2012.

Brian Hie et al. Efficient evolution of human antibodies from general protein language models. *Nature Biotechnology*, 42:275–283, 2024. doi: 10.1038/s41587-023-01763-2. URL https://doi.org/10.1038/s41587-023-01763-2.

Jingchao Jiang et al. Rapid in silico directed evolution by a protein language model. *Science*, 2024. doi: 10.1126/science.adr6006. URL https://doi.org/10.1126/science.adr6006.

Simon Kirjner, Jason Yim, Regina Barzilay, Tom Rainforth, Tommi Jaakkola, et al. Graph-guided sampling for discovering diverse AAV variants. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2307.00494.

Zeming Lin et al. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 2023. URL https://doi.org/10.1126/science.ade2574. Preprint and associated resources for ESM-2/ESMFold.

Michael S. Packer and David R. Liu. Methods for the directed evolution of proteins. *Nature Reviews Genetics*, 16(7):379–394, 2015. doi: 10.1038/nrg3927. URL https://doi.org/10.1038/nrg3927.

Saro Passaro, Gabriele Corso, Jeremy Wohlwend, Mateo Reveiz, Stephan Thaler, Vignesh Ram Somnath, Noah Getz, Tally Portnoi, Julien Roy, Hannes Stark, et al. Boltz-2: Towards accurate and efficient binding affinity prediction. *BioRxiv*, 2025.

Alexander Rives et al. 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. URL https://doi.org/10.1073/pnas.2016239118.

Karen S. Sarkisyan et al. 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.

Nataša Tagasovska, Vladimir Gligorijevic, Kyunghyun Cho, and Andreas Loukas. Implicitly guided design with propen: Match your data to follow the gradient. *Advances in Neural Information Processing Systems*, 37:35973–36001, 2024.

Jeremy Wohlwend, Gabriele Corso, Saro Passaro, Noah Getz, Mateo Reveiz, Ken Leidal, Wojtek Swiderski, Liam Atkinson, Tally Portnoi, Itamar Chinn, et al. Boltz-1 democratizing biomolecular interaction modeling. *BioRxiv*, pp. 2024–11, 2025.

Kevin Yang et al. Active learning-assisted directed evolution of proteins with low screening budgets. *Nature Communications*, 2025. doi: 10.1038/s41467-025-55987-8. URL https://doi.org/10.1038/s41467-025-55987-8.

Kevin K. Yang, Zhenming Wu, and Frances H. Arnold. 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/10.1038/s41592-019-0496-6.

A Related Work

Generative models. Protein language models and related generative approaches can model sequence distributions and generate protein sequences (Rives et al., 2021). More recent approaches include guided discrete diffusion for protein design (Gruver et al., 2023) and discrete flow models for protein co-design (Campbell et al., 2024). These methods are often evaluated in open-loop generation settings (i.e. *de novo* design and zero-shot tasks), where they generate a single sequence or a batch of diverse sequences. In contrast, MutaGen is constructed explicitly to operate inside an iterative, ranking-driven optimization loop.

Active / few-shot loops. Low-budget closed-loop optimization has been studied in the context of active learning-assisted directed evolution. Yang et al. (2025) examines different basic acquisition functions and surrogate models on large datasets, while Jiang et al. (2024) uses an ESM-2 based greedy ranking approach in a very low data active learning loop. MutaGen is complementary to these approaches: rather than using a calibrated fitness predictor to score candidates, it trains a generative transformation operator to propose candidates directly; importantly sidestepping the need for accurate absolute scores of protein samples, and instead relying soley on the relative ranks of samples.

Probabilistic sampling. Kirjner et al. (2024) diversify search through graph smoothing and energy-based MCMC. We similarly find that sampling-based methods can perform very well in realistic use cases. Our work rigorously compares the generative flow approach (MutaGen) against these probabilistic samplers (MCMC) to identify regimes where flow-matching offers superior sample efficiency.

Implicit-guidance, pair-based approaches. PropEn (Tagasovska et al., 2024) builds matched pairs (x, x^+) in which x^+ is experimentally better than x, thereby learning improvement directions without an explicit guidance potential or fitness oracle, similar to our pair-conditioned flow. At test time, PropEn performs implicit optimization by iteratively applying a trained encoder–decoder to a seed design, updating $x_{t+1} = f_\theta(x_t)$ until convergence to a fixed point, yielding a trajectory of progressively enhanced candidates (Tagasovska et al., 2024). MutaGen formalizes this intuition using the framework of Discrete Flow Matching and embedding it within an active learning cycle. While PropEn is presented in a general continuous-design formalism, its protein experiments focus on antibodies and adopt a representation that crucially relies on domain-specific alignment: sequences are one-hot encoded after alignment under a fixed antibody numbering scheme (AHo) produced by ANARCI (Dunbar & Deane, 2016), yielding a fixed-length positional grid on which per-site categorical modeling is well defined (Tagasovska et al., 2024). This canonical coordinate system is largely unique to antigen receptors and closely related families. This dependence on a family-specific numbering/alignment step makes straightforward extension of such discretizations to arbitrary proteins nontrivial, motivating MutaGen that is natively formulated over categorical sequence spaces without requiring a universal alignment scaffold.

B Experimental Setup

We mirror the closed-loop optimization workflow used in directed evolution (Arnold, 1998; Packer & Liu, 2015). Let Σ be the 20-letter amino-acid alphabet and L the sequence length. A protein sequence is $x \in \Sigma^L$ and its fitness is a scalar $f(x) \in \mathbb{R}$, obtained by measuring the variant (via an assay in the wet lab, or via a fixed predictor in *in silico* experiments).

Workflow. We run an iterative directed-evolution loop for T rounds. Each round follows the same pattern:

1. **Generate mutants:** each method proposes a batch of B mutant sequences based on the data collected so far.
2. **Measure fitness:** we evaluate each proposed mutant and record its fitness. In *in silico* experiments, this measurement is produced by a fixed predictor (an evaluation model called an oracle); in the wet lab, it is the assay readout.
3. **Update the method:** we add the new measurements to our running dataset and retrain any models used by the method (e.g., a surrogate regressor for EvolvePro and MCMC, or a mutant ranking for MutaGen).

Algorithm 1 MUTAGENGENERATE(x_0, θ) — one CTMC trajectory

Require: Parent $x_0 \in \Sigma^L$, learned parameters θ, step count N, mutable-site mask $\mathcal{M} \subseteq \{1, \dots, L\}$
Ensure: Mutant x_N

$x \leftarrow x_0$ ▷ current state
for $n \leftarrow 1$ **to** N **do**
  $t \leftarrow n/N$
  Compute logits $g_\theta^{(j,a)}(x, t)$ for all positions j and amino acids a
  $\pi_\theta^{(j)}(\cdot \mid x, t) \leftarrow \text{softmax}\big(g_\theta^{(j,\cdot)}\big)$
  for all $j \in \mathcal{M}$ **do** ▷ iterate over mutable residues
    Draw candidate $a \sim \pi_\theta^{(j)}$
    $R \leftarrow \max\big\{0,\ \pi_\theta^{(j)}(a) - \pi_\theta^{(j)}(x^{(j)})\big\}$ ▷ rate entry
    if $\text{Uniform}(0, 1) < R\,\Delta t$ **then**
      $x^{(j)} \leftarrow a$
    end if
  end for
end for
return x

Initialization. All *in silico* campaigns start from an identical first round of randomly generated mutants (across 20 seeds), reflecting a realistic setting without prior fitness information on the protein. For AAV, we restrict mutations to the standard 28-residue region used in prior benchmarks (Bryant et al., 2021); for GFP we allow mutations across the full sequence (Sarkisyan et al., 2016).

Oracle. For each protein benchmark (GFP, AAV), we train a separate predictor, typically referred to as an 'oracle', that provides fitness measurements during *in silico* optimization (we keep this model fixed throughout the closed-loop runs). Each sequence is embedded using the 8M-parameter ESM-2 model (Rives et al., 2021); we mean-pool the final hidden layer (320D) and feed the result into a 3-layer MLP that predicts a scalar fitness value. The predictor is trained with mean-squared error against the measured fitness values provided with the benchmark dataset.

Training Labels. We study a setting where absolute fitness measurements can be noisy or poorly calibrated, while their relative ordering is more reliable. After each round, we sort evaluated sequences by their measured fitness and derive pairwise preferences, so $x_i \succ x_j$ whenever the fitness measured for x_i is higher than for x_j (i.e., x_i ranks above x_j). MUTAGEN is trained only on such pairwise comparisons, not on the raw scalar fitness values.

By contrast, several baselines inherently require scalar fitness values (e.g., regression-based surrogates and MCMC-style acceptance rules). For these methods, we provide the scalar measurements, but compare to MUTAGEN which does not require access to this information and is therefore more robust to experimental noise.

C METHOD

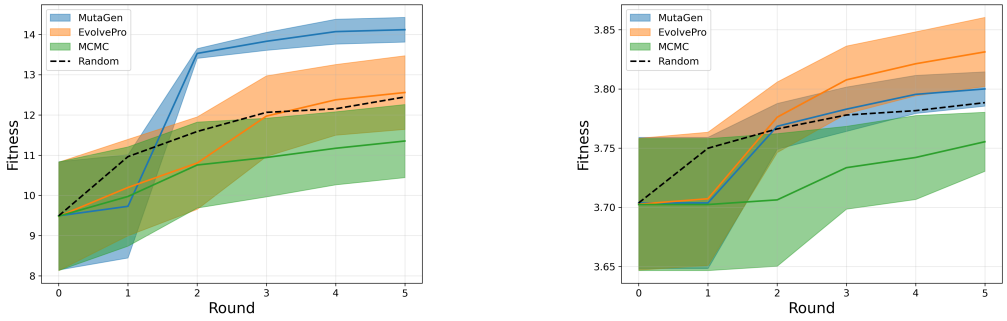
Pseudocode for MUTAGEN is shown in Algorithm 1.

D PERFORMANCE ON BENCHMARKS

Results shown below in Figure 3.

E ROUND SIZE TRADEOFF

We further investigate the tradeoff between allocating a fixed screening budget across many rounds with smaller batch sizes versus fewer rounds with larger batch sizes. This consideration is central for experimental design in protein optimization. Since the cost per screened mutant is typically



(a) AAV (b) GFP

Figure 3: Model performance across rounds on AAV (top) and GFP (bottom), reporting cumulative best fitness as a function of round. Solid curves show mean performance across 20 independent runs; shaded regions denote 95% confidence intervals for the mean. The dashed black curve denotes the random baseline.

Table 2: AAV optimization fitness (mean $\pm$ std) starting from random initialization. MUTAGEN outperforms baselines at low budgets($N = 20, 40, 60$), while EvolvePro achieves the best performance at higher budgets ($N = 80, 100$).

METHOD	$N = 20$	$N = 40$	$N = 60$	$N = 80$	$N = 100$
RANDOM	12.06 ± 1.52	12.06 ± 1.52	12.06 ± 1.52	12.06 ± 1.52	12.06 ± 1.52
EVOLVEPRO	13.72 ± 0.86	14.03 ± 0.69	14.23 ± 0.55	14.59 ± 0.46	14.72 ± 0.47
MCMC	12.22 ± 1.49	12.43 ± 1.03	12.76 ± 0.87	12.96 ± 0.94	12.54 ± 1.15
MUTAGEN	$\mathbf{14.26 \pm 0.66}$	$\mathbf{14.27 \pm 0.67}$	$\mathbf{14.46 \pm 0.69}$	14.36 ± 0.72	14.24 ± 0.71

constant, each additional round of evolution substantially increases the overall duration and logistical complexity of an optimization campaign.

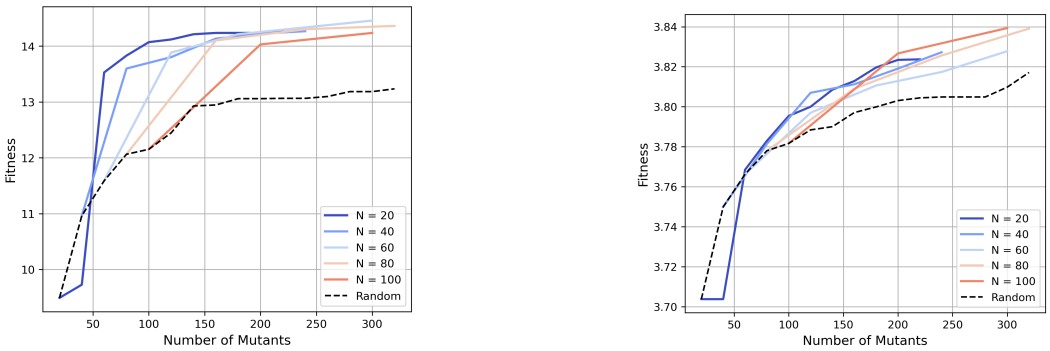
To study this tradeoff, we benchmark MUTAGEN on both GFP and AAV using evolution campaigns composed of multiple rounds with batch sizes ranging from 20 to 100 mutants per round. On AAV (Figure 4a), we observe that distributing the budget across more rounds with smaller batch sizes enables MUTAGEN to identify high-fitness variants more rapidly. This behavior is consistent with classical active learning dynamics, where more frequent model updates allow for increasingly informed selection of subsequent mutants.

In contrast, results on GFP (Figure 4b) reveal a markedly different regime: fitness improvements achieved by MUTAGEN are largely insensitive to how the screening budget is distributed across rounds, provided that the total number of observed mutants is held constant. That is, splitting a fixed budget across multiple rounds offers little advantage over screening the same number of mutants in fewer rounds.

Taken together, these results suggest that MUTAGEN can exhibit both standard active learning behavior—where performance improves with more rounds—and regimes in which the number of rounds can be substantially reduced without sacrificing performance. The latter case is particularly appealing for experimental practitioners, as it implies that, for some proteins, optimization campaigns using MUTAGEN can be completed in significantly less time by concentrating screening into fewer rounds. Detailed numerical results corresponding to Figures 4a and 4b are provided in Table 2 and 3. Results shown below in Figure 4a and 4b.

F GFP NOISE ROBUSTNESS

Results shown below in Figure 5.



(a) AAV (b) GFP

Figure 4: Tradeoff between running more rounds with fewer mutants per round versus fewer rounds with more mutants per round for MUTAGEN. The x-axis denotes the cumulative number of screened mutants across rounds, and the y-axis reports the best-achieved fitness. Curves correspond to different per-round screening budgets $N \in \{20, 40, 60, 80, 100\}$, where N is the number of sequences tested in each round.

Table 3: GFP optimization fitness (mean $\pm$ std) starting from random initialization. All methods achieve comparable performance on GFP, with differences within measurement noise.

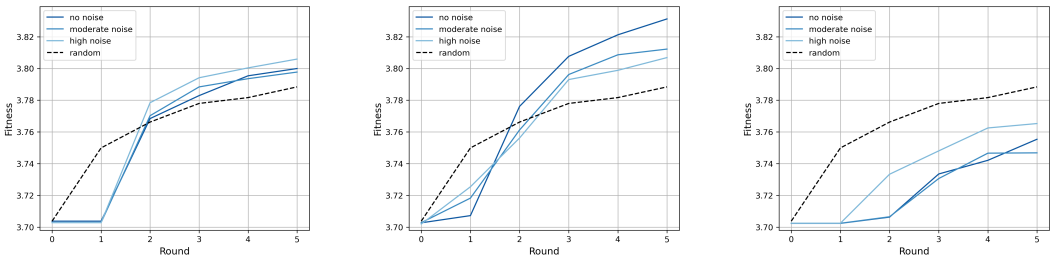
METHOD	$N = 20$	$N = 40$	$N = 60$	$N = 80$	$N = 100$
RANDOM	3.78 ± 0.04	3.78 ± 0.04	3.78 ± 0.04	3.78 ± 0.04	3.78 ± 0.04
EVOLVEPRO	3.88 ± 0.06	3.85 ± 0.06	3.86 ± 0.05	3.89 ± 0.06	3.89 ± 0.06
MCMC	3.79 ± 0.04	3.82 ± 0.07	3.83 ± 0.04	3.83 ± 0.03	3.83 ± 0.05
MUTAGEN	3.82 ± 0.05	3.83 ± 0.03	3.83 ± 0.02	3.84 ± 0.03	3.84 ± 0.03

G EXPERIMENTAL DETAILS

G.1 HIGH-THROUGHPUT CLONING OF NANOLUC LUCIFERASE MUTANTS

Briefly, each optimization round consisted of constructing a library of NanoLuc sequence variants, transient expression of variants in HEK293FT cells, and quantitative measurements of luminescence output using a plate-based assay. For each variant, we measured raw bioluminescence intensity with three technical replicates. We performed plate-specific background subtraction by first averaging technical replicates and then subtracting the corresponding plate background value, estimated from three no-media wells on the same plate. To make measurements comparable across plates and experimental days, we normalized luminescence using internal control mutants and the wild-type (WT); we report fold-change relative to the WT from a reference experiment.

Plasmid expression constructs encoding NanoLuc luciferase mutants were generated using the NEBuilder® HiFi DNA Assembly Master Mix (NEB, E2621S) following manufacturer's manual. The Nanoluc plasmid backbone was digested with FastDigest NheI (Thermo Fisher Scientific, FD0974) and XbaI (Thermo Fisher Scientific, FD0684) at 37oC for 1 hr. The linearized vector was then purified with the Monarch® DNA Gel Extraction Kit (NEB, T1120L) and eluted in UltraPure DNase/RNase-Free Distilled Water (Thermo Fisher Scientific, 10977023). NanoLuc luciferase mutant inserts, flanked by 20-bp Gibson homology arms, were synthesized as eBlock® dsDNA fragments (IDT) and assembled into the digested vector by Gibson assembly at 50oC for 30 minutes. Following assembly, 1.5 µL of each reaction was transformed into 25 µL of chemically competent Stbl3, generated using the Mix & Go! E.coli Transformation Kit (Zymo Research, T3001) from One Shot™ Chemically Competent E. coli (Thermo Fisher Scientific, C737303). Transformations were performed by heat shock at 42oC for 30 s, followed by plating on LB agar plates with 100 µg/mL ampicillin and overnight incubation at 37oC. Individual colonies were picked and cultured in 1.5 mL Terrific Broth (Fisher Scientific, A1374301) in 96-well, 2-mL plates at 37oC with shaking at 225 rpm. After overnight growth, plasmid DNA was extracted using the ZymoPURE Plasmid Purification



(a) MutaGen, GFP (b) EvolvePro, GFP (c) MCMC, GFP

Figure 5: GFP robustness to noisy fitness measurements. Closed-loop optimization trajectories under additive noise in the feedback signal (no / moderate / high), with the random baseline shown as a dashed line. Columns correspond to MUTAGEN (left), EvolvePro (middle), and MCMC (right). Curves show the mean cumulative best fitness over 20 runs.

Kit buffers and 96-well miniprep filters following manufacturer's manual. Plasmid concentration and purity (A260/A280) were assessed using a NanoDrop One spectrophotometer (Thermo Fisher Scientific). All constructs yielded concentrations $\geq$ 10 ng/µL with A260/A280 ratios between 1.8 and 2.0.

G.2 MAMMALIAN CELL CULTURE AND TRANSFECTION

HEK293FT cells (Thermo Fisher Scientific, R70007) were cultured in DMEM/Glutamax (Thermo Fisher Scientific, 10569044) supplemented with 10% fetal bovine serum (Thermo Fisher Scientific, A5256801) and 1x penicillin-streptomycin (Thermo Fisher Scientific, 15140122) at 37oC, 5% CO2 in a humidified incubator. For transfection, cells were seeded at a density of 15,000 cells per well in 96-well plates 16-20 h prior to transfection to achieve 80–90% confluency. Transfection mixtures were prepared using Lipofectamine™ 3000 (Thermo Fisher Scientific, L3000008) by combining 50 ng plasmid DNA, 0.2 µL P3000™ reagent, and 0.2 µL Lipofectamine 3000 in Opti-MEM™ reduced-serum medium (Thermo Fisher Scientific, 31985070) to a final volume of 10 µL. The mixtures were incubated at room temperature for 15 min and then added dropwise to the cells.

G.3 NANO-GLO®LUCIFERASE ASSAY

NanoLuc luciferase activity was measured two days after transfection using the Nano-Glo® Luciferase Assay System (Promega, N1120) following manufacturer's manual. Briefly, the Nano-Glo assay reagent was prepared by mixing Nano-Glo luciferase substrate, Nano-Glo lysis buffer, and PBS at a ratio of 1:50:200. An equal volume of assay reagent was added directly to each well, and plates were incubated for 10 min at room temperature in the dark. Luminescence measurements were performed by transferring 100 µL of lysed samples to white, opaque 96-well plates, followed by signal detection using a BioTek Synergy™ 4 plate reader.

G.4 PLATE AND DAY NORMALIZATION

To make measurements comparable across plates and experimental days, we applied a two-stage calibration. First, within the new experiment, we fit a linear transformation on log-intensity values using the three internal control mutants and the wild-type (WT) measured on Plate 1, and applied this transformation (log transform, linear map, then back-transform by $10^{(\cdot)}$) to all wells to align plates within the new experiment. Second, to align the new experiment to a reference experiment (collected in a separate run that includes the same internal control mutants), we fit an analogous linear transformation on log-intensities using the same three internal control mutants plus the Plate-1 WT from the new experiment and the reference experiment, and applied it to all transformed values. Finally, we report fold-change relative to the WT raw intensity from the reference experiment.

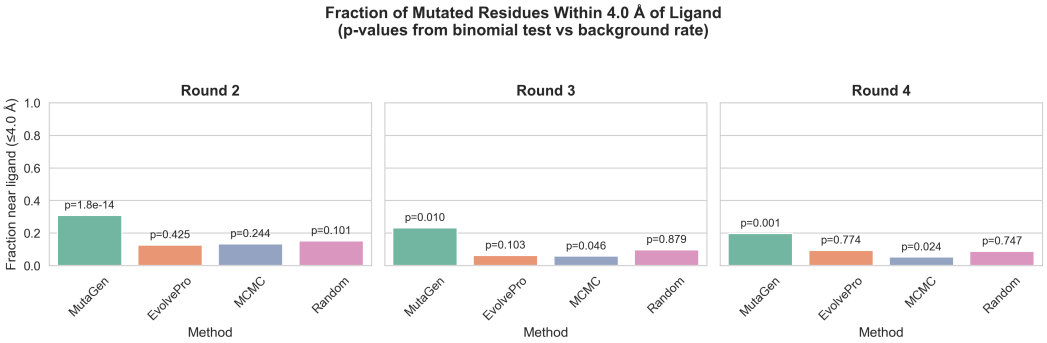


Figure 6: MUTAGEN mutates significantly more residues that are in close proximity to the furimazine ligand compared to a random mutation baseline. EvolvePro, MCMC and the random mutation strategy do not have this enrichment in their mutations across the four round experimental campaign.

G.5 ACTIVE SITE ENRICHMENT NANOLUC CAMPAIGN

To better understand the mechanistic basis of MUTAGEN's performance in the NanoLuc wet-lab campaign, we analyzed whether the mutations proposed across rounds preferentially target residues proximal to the substrate binding site. Importantly, MUTAGEN is not provided with any structural information during training or inference; all proposals are generated purely from ranked sequence comparisons and a fixed, sequence-based saliency prior.

For each method and each round, we computed the fraction of mutated residues that lie within 4 Å of the furimazine ligand in the NanoLuc crystal structure. This radius was chosen to capture residues directly involved in substrate binding or catalytic modulation. We compare these fractions against a background rate defined by uniform random mutation over the same set of mutable positions, and assess statistical significance using a binomial test.

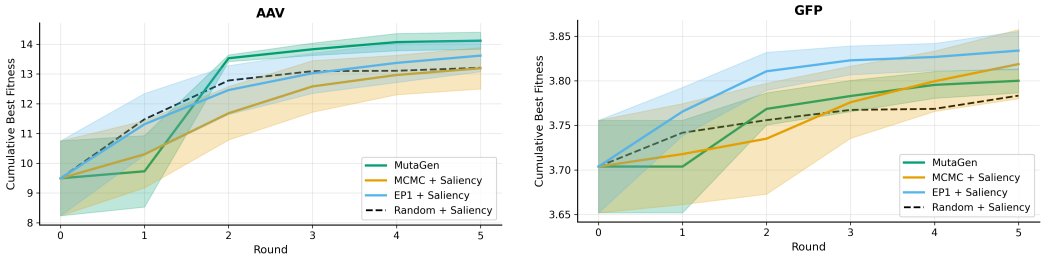
As shown in Fig. 5, MUTAGEN exhibits a significant enrichment of mutations near the active site beginning in round 2 and persisting through rounds 3 and 4 (e.g., $p < 0.05$ in all later rounds). In contrast, EvolvePro, MCMC, and the random mutation baseline do not show consistent or significant enrichment over the background rate at any stage of the campaign. Visual inspection of mutation patterns mapped onto the NanoLuc structure further corroborates this finding: MUTAGEN mutations cluster around the furimazine binding pocket, whereas baseline methods distribute mutations broadly across the protein surface.

These results indicate that preference-based discrete flow matching can implicitly recover functionally important, substrate-proximal regions purely from ranked experimental feedback.

H SALIENCY-MAP ABLATIONS FOR BASELINES

Our implementation of MUTAGEN optionally restricts mutable positions using an attention-based saliency mask computed once on the wild-type sequence (a simple, unsupervised prior over likely-functional sites). To isolate whether the gains we observe stem primarily from this positional prior versus the underlying optimization algorithm, we perform an ablation in which we apply the same saliency mask to all baselines by restricting their proposed mutations to the identical set of allowed positions.

Figure 7 shows that adding the saliency mask can yield small improvements for some baselines, but does not qualitatively change the conclusions: relative method ordering and the main gap on AAV remain largely unchanged, and differences on GFP remain modest. In particular, MUTAGEN retains a clear advantage on the more challenging AAV landscape even when all methods are granted the same mutation-location prior. We emphasize that the published versions of these baselines do not incorporate saliency-based mutation masks; we include this experiment solely as a controlled comparison to quantify the effect of this auxiliary prior.



(a) AAV (b) GFP

Figure 7: Saliency-mask ablation across methods. We apply the same attention-derived saliency mask (used by MUTAGEN) to all baselines by restricting mutations to the identical set of allowed positions. Curves report mean cumulative best fitness over 20 runs with 95% CIs; the dashed curve denotes the random baseline with the same saliency constraint. Applying saliency yields at most minor changes for baselines and does not alter the main conclusions.

I DISCRETE FLOW MATCHING HYPERPARAMETERS

Table 4 summarizes the hyperparameters used for the Discrete Flow Matching (DFM) directed evolution experiments.

Table 4: Hyperparameters for DFM-based directed evolution with saliency-guided mutation selection.

Category	**Parameter**	**GFP**	**AAV**
Evolution	Rounds	5	5
	Samples per round	20	20
Saliency	Edit budget	15	8
	Increase per round	0	0
	Temperature (τ)	1.0	1.0
Sampler Model	Encoder	ESM-2 (8M)	
	MLP hidden layers	$1024 \to 512 \to 68$	
Training	Learning rate	0.001	
	Optimizer	Adam	
	Epochs per round	2	
	Sampling steps	10	

J MCMC HYPERPARAMETERS

Table 5 summarizes the hyperparameters used for the MCMC-based directed evolution experiments.

K EVOLVEPRO HYPERPARAMETERS

Table 6 summarizes the hyperparameters used for EvolvePro directed evolution experiments.

Table 5: Hyperparameters for MCMC-based directed evolution.

Category	Parameter	GFP	AAV
Evolution	Rounds	5	5
	Samples per round	20	20
	Number of branches	200	200
	Steps per branch	30	16
Mutation	Edit budget	15	8
	Initial edit limit	1	1
	Mutation type	Single point	
Embedding	Encoder	ESM-2 (8M)	
	Embedding dimension	320	
Top Layer	MLP hidden layers	$256 \rightarrow 128$	
	Dropout	0.3	
	Learning rate	0.005	
	Optimizer	Adam	
	Epochs per round	50	

Table 6: Hyperparameters for EvolvePro-based directed evolution.

Category	Parameter	GFP	AAV
Evolution	Rounds	5	5
	Samples per round	20	20
	Candidate pool size	100,000	100,000
Mutation	Edit budget	15	8
	Mutation type	Random multi-point	
	Selection strategy	Greedy ranking	
Embedding	Encoder	ESM-2 (8M)	
	Embedding dimension	320	
Predictor	Model	Random Forest	
	Number of estimators	100	
	Max depth	10	
	Max features	0.5	
	Criterion	Friedman MSE	