How to make the most of your masked language model for protein engineering

Calvin McCarter
BigHat Biosciences*

Nick Bhattacharya
BigHat Biosciences

Sebastian W. Ober
BigHat Biosciences

Hunter Elliott
BigHat Biosciences
helliott@bighatbio.com

Abstract

A plethora of protein language models have been released in recent years. Yet comparatively little work has addressed how to best sample from them to optimize desired biological properties. We fill this gap by proposing a flexible, effective sampling method for masked language models (MLMs), and by systematically evaluating models and methods both in silico and in vitro on actual antibody therapeutics campaigns. Firstly, we propose sampling with stochastic beam search, exploiting the fact that MLMs are remarkably efficient at evaluating the pseudo-perplexity of the entire 1-edit neighborhood of a sequence. Reframing generation in terms of entire-sequence evaluation improves the quality of proposed sequence variants, and enables flexible guidance with multiple optimization objectives. Secondly, we report results from our extensive in vitro head-to-head evaluation for the antibody engineering setting, comprising multiple models and samplers. This reveals that the choice of sampling method can have a substantial impact, motivating future research into this under-explored area.

1 Introduction

Antibodies are a class of natural immune proteins that have served as the starting point for the most successful class of biologic drugs [Carter and Lazar, 2018], and are thus a focus of efforts to utilize machine learning to accelerate drug discovery. Developing an antibody therapeutic typically involves first identifying a candidate via animal immunization [Köhler and Milstein, 1975], *in vitro* screening libraries [McCafferty et al., 1990], or *in silico* "de novo" design methods [Bennett et al., 2024]. Second, because candidates usually require substantial improvement, *in vitro* iterative optimization is used to refine the candidate into a valid therapeutic [Lu et al., 2020].

The space of possible antibody sequence mutations is combinatorially vast, and only a small number of mutations are typically advantageous; meanwhile, multi-objective optimization experiments are limited to hundreds of sequences and are costly and slow [Buchanan et al., 2025]. Machine learning-guided iterative design improves this process by prioritizing mutated sequences which are likely to yield improvements along desired criteria [Ober et al., 2025, Kuo et al., 2026, Rao et al., 2026]. In each round, a language model receives top-performing candidates (seeds), and optionally receives additional sequence scoring functions (such as supervised models trained on sequences previously measured in the lab). The generative model, combined with a sampling algorithm, outputs mutated sequences, some of which are selected for the next round of lab measurements.

*Work done while at BigHat Biosciences. Now at Aureka Biotechnologies. Correspondence to: mccarter.calvin@gmail.com.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

Many protein language models [Madani et al., 2020, Lin et al., 2022, Alamdari et al., 2023, Hayes et al., 2025, Bhatnagar et al., 2025] and antibody-specific language models [Shuai et al., 2021, Olsen et al., 2024, Fournier et al., 2024, Turnbull et al., 2024, Burbach and Briney, 2025] have been released in recent years. Because protein sequences lack causal directionality, and because iterative design frequently involves conditioning on arbitrary fixed subsequences, masked language models (MLMs) rather than causal language models (CLMs) are the typical choice. Depending on the goals of the optimization campaign, the language model chosen may be one trained on a large human antibody dataset [The Observed Antibody Space, OAS; Kovaltsuk et al., 2018], on a database of antibodies with known structures [enriched for therapeutic antibodies, Dunbar et al., 2014], or on a more generic protein sequence database [Consortium, 2015]. Regardless of the choice of model, the premise is that model likelihoods will positively correlate with the probability of the sequence yielding a valid, functional antibody.

The goal of a sampling algorithm is to elicit from the MLM a large, diverse set of sequences with relatively high model likelihood and some notion of proximity to the seed sequences. Unfortunately, systematic evaluation of sampling MLM algorithms is essentially non-existent. Furthermore, previously-adopted MLM sampling algorithms employ mutation-centric iterative sampling procedures, essentially any-order analogues of greedy sampling methods for causal language model generation [Hoogeboom et al., 2021, Sahoo et al., 2024]. These methods are computationally costly and (as we will show later) empirically tend to elicit unlikely, dysfunctional sequences.

It is also frequently desirable to bias the language model's distribution based on additional scoring functions. Sometimes these are differentiable functions, such as neural networks trained on binding affinity measurements. In other cases these scores, such as the OASis percentile risk score for immunogenic adverse events [Prihoda et al., 2022] or isoelectric point (predictive of stability and pharmacokinetics quality), are both non-differentiable and require as input a clean (rather than partially-masked) sequence. Mutation-centric sampling methods poorly handle such scenarios without additional approximations [Raghu et al., 2025] or costly computations [Tang et al., 2025].

In this work, we propose to instead take a sequence-centric approach, where we evaluate full sequences according to the MLM, and search sequence space starting from the seed. We show that this is surprisingly computationally efficient and effective at producing sequences preferred by the MLM and optional scoring functions. Next, we evaluate a variety of models and algorithms, first *in silico* and then *in vitro* on actual antibody therapeutics programs. We find that the choice of sampler has a surprisingly large impact, making it a consequential and under-explored design choice alongside model selection. We also report other interesting findings, such as the fact that ESM-2 [Lin et al., 2022], though trained on generic protein sequences, is still very effective for antibody optimization.

1.1 Related Work

CLM sampling algorithms Sampling algorithms for causal language models (CLMs) currently tend to follow the greedy decoding paradigm, making decisions solely based on the distribution for the next token. Argmax decoding often produces repetitive text, so methods like softmax temperature annealing [Ackley et al., 1985, Kirkpatrick et al., 1983] and nucleus (top-p) filtering [Holtzman et al., 2019] are widely used to spread out probability mass among the most likely tokens, while eliminating the least likely tokens. Methods based on (partial) sequence likelihoods, such as beam search [Gimpel et al., 2013, Meister et al., 2020] and stochastic beam search (SBS) [Kool et al., 2019] have been proposed, but are less frequently used. These approaches have complexity $O(L^2)$, where L is the sequence length.

MLM sampling algorithms Sampling from masked language models (MLMs) for protein engineering has a very different design space from that of text CLMs. The chief differences are the lack of a pre-defined left-to-right order, the notion of a seed sequence, and the desired proposal of a diverse batch for subsequent *in silico* filtering and finally *in vitro* measurement. Also, sometimes specific regions are constrained to be unchanged, such as the complementarity-determining region (CDR) if binding is satisfactory, or the framework region (FR) if only binding requires further modification.

The dominant paradigm for iterative refinement generation from MLMs (and discrete diffusion models), both in NLP [Lee et al., 2018, Ghazvininejad et al., 2019, Hoogeboom et al., 2021] and protein

engineering [Hayes et al., 2025, Hopf et al., 2026] is mutation-centric: iteratively mask and resample subsets of positions. Denoising sampling [Hoogeboom et al., 2021] is the closest analogue to CLM greedy decoding, and was adopted for protein optimization by Hayes et al. [2025]. To edit up to $E \leq L$ residues, conditioned on the remaining $L - E$ fixed positions, it first masks all E positions then runs unmasking for E iterations. Alternatively, Gibbs sampling [Geman and Geman, 1984, Ackley et al., 1985] has been adopted for protein [Johnson et al., 2021] engineering, and tends to propose sequences which are closer to the seed sequence. Gibbs-like sampling masks-and-unmasks for E iterations; at each iteration 1 position is masked, while the remaining $L-1$ positions are unmasked. For both autoregressive denoising sampling and Gibbs-like sampling, the sampling step can be replaced with an argmax; this was previously employed for antibody humanization [Gordon et al., 2024]. Mutation-centric samplers also require a strategy to choose the next position to decode; for example, the ESM-3 work [Hayes et al., 2025] optimized proteins via selection by either lowest entropy or max probability.

Mutation-centric MLM samplers have complexity $O(EL^3)$ per generated sequence. To obtain N unique sequences, one must run the sampling method at least N times; independent runs are not guaranteed to produce unique sequences.

MLM guidance with scoring functions Prior work on incorporation of additional scoring functions is also limited. The ESM-3 work [Hayes et al., 2025] proposes using derivative-free guidance [Li et al., 2024] but requires scoring functions to accept unclean (partially-masked) sequences. Discrete Guidance [Nisonoff et al., 2024] similarly requires the predictor to score partially-masked sequences. Alternatively, Raghu et al. [2025] does not force scoring functions to evaluate unclean sequences, but to remain compatible with mutation-centric sampling, assumes that the scoring function is a product of experts over positions in the sequence [Schiff et al., 2024]. One recent work [Tang et al., 2025] uses Monte Carlo tree search (MCTS) to avoid these constraints by performing a complex search and sampling process that scores many clean sequences to estimate the value of one partially-masked sequence; this is problematic for scores like OASis percentile which are very expensive.

Benchmarking Benchmarking on sampling for protein optimization is scarce. The most extensive sampler benchmarking results for protein optimization have been reported in [Darmawan et al., 2025]. However, it analyzed sampling methods for a causal protein language model [Notin et al., 2022], and it involved only *in silico* evaluation. The argmax versions of denoising and Gibbs-like sampling were evaluated *in vitro* in [Gordon et al., 2024]; argmax Gibbs was marginally better.

Code is available at `https://github.com/calvinmccarter/mlm4proteins`.

2 Methods

2.1 Background on protein MLMs

Denote the set of 20 amino acids by $\mathcal{A} = \{a_1, a_2, ..., a_{20}\}$. Let $\boldsymbol{x} = [x_1, ..., x_L] \in \mathcal{A}^L$ denote a protein sequence, representing a sequence of L amino acids with each $x_i \in \mathcal{A}$. A masked language model (MLM) takes an input sequence $\boldsymbol{x}$ and outputs an $L \times 20$ matrix $\boldsymbol{Y} = [\boldsymbol{y}_1, \ldots, \boldsymbol{y}_L]^\top$. Each $\boldsymbol{y}_i$ can naively be interpreted as approximate log-probabilities over the set of amino acids, with the "probability" for each residue at location i given by

$$\hat{p}(x_i = a_r | \boldsymbol{x}_{j \neq i}) \approx \frac{\exp(\boldsymbol{Y}_{ir}/\tau)}{\sum_{k=1}^{20} \exp(\boldsymbol{Y}_{ik}/\tau)} = \texttt{softmax}_\tau(\boldsymbol{Y}_{i,:}), \tag{1}$$

where $\tau > 0$ is the temperature of the softmax function. Note that $\boldsymbol{Y}$ was computed from the full sequence $\boldsymbol{x}$ including the value of x_i, so this only approximately holds. This *wildtype-marginal* approximation [Meier et al., 2021] allows a single $O(L^2)$ forward-pass[2], by replacing $\hat{p}(x_i = a_r | \boldsymbol{x}_{j \neq i})$ with $\hat{p}(x_i = a_r | \boldsymbol{x})$.

To obtain the true conditional probability distribution $\hat{p}(x_i | \boldsymbol{x}_{j \neq i})$, one instead masks x_i such that $\tilde{\boldsymbol{x}}^{(i)} = [x_1, ..., x_{i-1}, \texttt{MASK}, x_{i+1}, ..., x_L]$, and computes MLM output $\tilde{\boldsymbol{Y}}^{(i)}$, thus obtaining

$$\hat{p}(x_i = a_r | \boldsymbol{x}_{j \neq i}) = \texttt{softmax}_\tau(\tilde{\boldsymbol{Y}}^{(i)}_{i,:}). \tag{2}$$

[2]We assume that MLMs use quadratic complexity scaled dot-product attention.

A separate forward-pass is needed for each i, and each time $L-1$ rows of $\boldsymbol{Y}$ are discarded. Thus each iteration of mutation-centric sampling has cost $O(L^3)$.

Unlike CLMs, MLMs do not offer an efficient way to compute a sequence log-likelihood or perplexity. Instead, a pseudo-log-likelihood (PLL) is computed by summing over positions $\sum_i \log \hat{p}(x_i = a_r|\boldsymbol{x}_{j\neq i})$, for a total cost of $O(L^3)$ [Salazar et al., 2020, Wang and Cho, 2019]; in contrast, the CLM true log-likelihood (LL) costs only $O(L^2)$.

2.2 Sequence-centric search via MLM pseudo-log-likelihoods

Instead of asking the MLM to generate mutations, here we propose to ask the MLM to evaluate sequences via its PLL, which allows us to reduce our task to a search problem. This would appear to be infeasible given the cost of computing the PLL, but it turns out that having computed the PLL for a single sequence, the (approximate) PLLs for all sequences that are one substitution away come for free. Let $\boldsymbol{x}'$ be a sequence that is one substitution away from template $\boldsymbol{x}$ at position k. We then use an approximation to the PLL given by

$$PLL(\boldsymbol{x}';\tau) \approx \sum_{i=1}^{L} \log\left(\texttt{softmax}_\tau(\tilde{\boldsymbol{Y}}^{(i)}_{i,\texttt{index}(x'_i)})\right) \tag{3}$$

where we use the exact conditional probabilities at position k, while using the template's conditional probabilities $\hat{p}(x_i|\mathbf{x}_{j\neq i})$ elsewhere. This latter approximation is known as the *masked-marginal approximation*, widely used for zero-shot mutation effect prediction [Meier et al., 2021], including on ProteinGym [Notin et al., 2023], due to its strong performance.

The ability to quickly compute approximate PLLs motivates running beam search for E steps, starting from the initial seed sequence to naturally enforce proximity. Each time we expand a sequence from the beam, it becomes the new template sequence $\boldsymbol{x}$, and we evaluate its PLL, thus limiting the approximation error. With a beam of size B, we incur a cost of $O(BEL^3)$, obtaining $20BEL$ unique sequences. For a realistic scenario where $E = 5, L = 100$ and one desires a large number of sequences pre-filtering, the $20EL\times$ speedup compared to mutation-centric sampling is enormous.

As we will see later, this approach tends to elicit higher-quality sequences from MLMs than mutation-centric sampling. Recall that at each beam expansion, we recompute the exact stepwise-masked conditionals of the new template, so the masked-marginal approximation error never compounds across the E steps. Moreover, while mutation-centric methods commit irreversibly to each residue based on a local conditional, beam search scores whole sequences and maintains B alternatives at every step. Beam search thus lets locally-attractive but globally-incoherent paths die out, in favor of sequences the MLM endorses globally.

In practice, we would also like to obtain a diverse set of sequences. We adapt stochastic beam search [Kool et al., 2019], which adds Gumbel noise before ranking [Gumbel, 1954, Maddison et al., 2014] to balance sequence likelihood and diversity. The aforementioned softmax temperature, because it scales the likelihood term but not the Gumbel term, adjusts the balance. Note that while Kool et al. proposed perturbing log-probabilities so that ranking by perturbed scores is equivalent to sampling without replacement [Vieira, 2014], their work considered beam search for autoregressive generation. In addition to our different approximate scores, our search setting is structurally different: theirs is rooted at the empty sequence with partial-sequence nodes and branching factor $|A|$; ours is rooted at the seed with full-sequence nodes and branching factor $(|A|-1)L$.

Our approach can be applied to CLM LLs instead of MLM PLLs, if one wants to use a protein or antibody CLM to generate sequences near to a seed sequence, rather than to generate sequences from scratch. However, this requires $20L$ separate LL calculations, for a net $20\times$ cost increase.

We verified the PLL approximation empirically on Trastuzumab, finding strong agreement with the exact PLL over all 190 one-edit CDR-H3 variants across four models, shown in Table 1. The approximation alone correlates strongly with the true PLL, while adding Gumbel noise for stochastic ranking reduces these correlations as expected. We extended the above analysis to the larger 7 antibody panel from Hie et al. [2024]; the results further confirm the strength of the approximation, and are provided in Appendix A.1. A deeper theoretical discussion of PLL approximations in beam search is given in Appendix A.2.

	Approximation		Approximation + Gumbel	
Model	Pearson	Spearman	Pearson	Spearman
ESM2-650M	0.964	0.940	0.735	0.727
AbLang2	0.869	0.869	0.752	0.729
ProteinMPNN	0.944	0.934	0.721	0.724
AbMPNN	0.927	0.850	0.726	0.615

Table 1: Approximate-vs-true PLL correlation over all 190 one-edit CDR-H3 variants of Trastuzumab ($\tau = 1$).

2.3 Multi-objective optimization (MOO) with gradient-free guidance

Our proposed search framework treats both the MLM and optional scoring functions as black boxes, neither providing partially-masked sequences nor requesting gradients. Any multi-objective scalarization technique is thus applicable for our framework. Recent works addressing protein multi-objective optimization have employed Pareto non-dominated sorting (NDS) [Chen et al., 2025, Tang et al., 2025, Zhang and Chatterjee, 2025]. We consider that, as well as smooth Tchebycheff scalarization (STS) [Lin et al., 2024], which attempts to improve the performance of all objectives, rather than any objective as does NDS. NDS does not admit weights over objectives; in some of our experiments, we employed a weighted extension of STS. See Appendix A.3 for more details.

3 Experiments

3.1 *In silico* method evaluation on Trastuzumab CDR-H3 optimization

We evaluated our proposed method using Trastuzumab's VH domain; we diversify the 10-residue CDR-H3 with $E \in \{1, \ldots, 7\}$ edits and evaluate generators under ESM2-650M [Lin et al., 2022], AbLang2 [Olsen et al., 2024], ProteinMPNN [Dauparas et al., 2022], and AbMPNN [Dreyer et al., 2023]. For evaluation, we retrained a CNN classification oracle on the 367k-sequence binding dataset of Chinery et al. [2024] (reproducing its near-perfect reported test AUROC, at 0.995); secondarily we evaluate humanness via OASis percentile [Prihoda et al., 2022]. For guided generation, we also retrained a CNN classification model (dubbed "`MasonCNN`" hereafter) on the 34k-sequence binding dataset of Mason et al. [2021] (reproducing its reported 0.91 test AUROC, at 0.914). In addition to three unguided methods (denoising, Gibbs, and our beam search), we evaluate a Gibbs with `MasonCNN` PoE guidance baseline, our beam search with `MasonCNN` STS guidance, and our beam search with equally-weighted `MasonCNN` and OASis percentile guidance. Further experimental details are given in Appendix A.4.

Results for ESM2 are shown in Figure 1. Under ESM2, our unguided beam search dominates the mutation-centric baselines on binding (left panel): at six realized edits, mean `p(binder)` ≈ 0.95, versus ≈ 0.43 for denoising and ≈ 0.44 for Gibbs. ESM2's antibody priors are strong enough on their own that adding `MasonCNN` guidance does not further improve binding for our approach, but pairing it with OASis humanness as a second objective reaches the Pareto-optimal upper-right region at both $E = 3$ and $E = 6$ (middle panel), raising humanness from the Trastuzumab-seed while preserving high `p(binder)`, and dominating every Gibbs or denoising-based method. We observed that `MasonCNN` guidance did not improve binding for Gibbs; Gibbs PoE guidance with the strong oracle showed somewhat improved binding probability, but still worse binding than even our unguided beam search (left panel). We observed somewhat less pairwise sequence diversity among generated variants passing basic quality filters (bottom right panel), with Beam + STS (`MasonCNN` + OASis) the most concentrated: multi-objective guidance trades a controlled amount of diversity for higher reliability.

Results for AbLang2, ProteinMPNN, and AbMPNN, are qualitatively similar, and are presented in Appendix A.5. Notably, we find that AbLang2 likelihoods are negatively correlated with `p(binder)`, necessitating supervised guidance; ProteinMPNN and AbMPNN have performance strongly dependent on the availability of antigen structure; and that ESM2-650M is at least as good as the inverse-folding MPNNs even with the antigen.

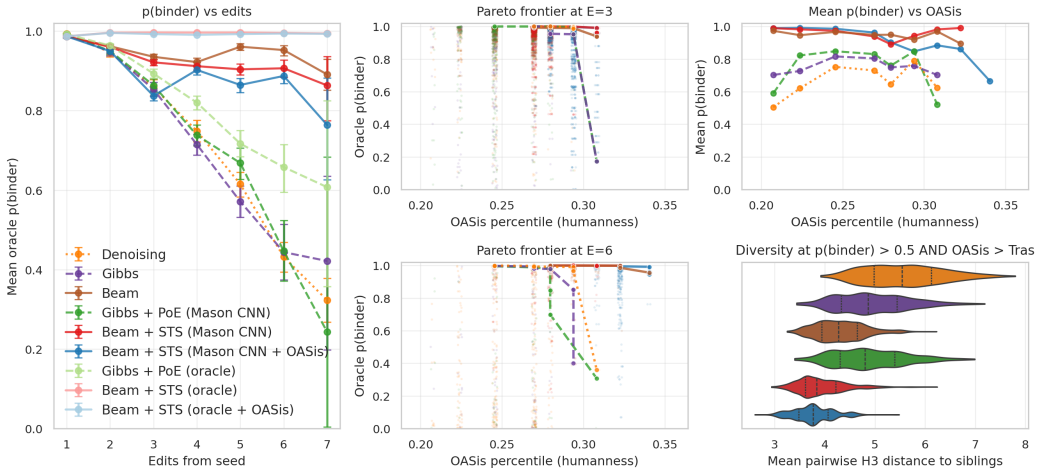


Figure 1: ESM2-650M on the Trastuzumab CDR-H3 *in silico* evaluation. Color groups `MasonCNN`-guided methods with their lighter-shaded strong-oracle counterparts; linestyle indicates methodology (solid: ours; dashed: previous). **[Left:]** Mean predicted `p(binder)` vs edits from the seed; error bars depict 95%-bootstrap CIs. **[Middle:]** OASis × `p(binder)` Pareto frontiers. Faded dots show the underlying generated variants. **[Top Right:]** Mean `p(binder)` at each OASis percentile bucket, pooling across all edits. **[Bottom right:]** Pairwise H3-diversity violins restricted to "high-quality" variants — those with `p(binder)` > 0.5 AND OASis > 0.245, the Trastuzumab seed.

We also use this setting to evaluate the behavior of our approach as a function of the beam size and temperature hyperparameters, with results in Appendix A.6. These results show that quality is very robust to the choice of hyperparameters, and that beam size could have been increased to larger values than used in our experiments, increasing diversity without decreasing quality. Furthermore, using this setting, we empirically demonstrate the substantial computational speedup from our approach, with results in Appendix TODO.

3.2 *In silico* benchmarking

We then evaluated a wide variety of combinations of models and sampling algorithms *in silico*. We evaluated nine MLMs: ESM-2 (35M, 150M, and 650M params) [Lin et al., 2022], Sapiens (500K) [Prihoda et al., 2022], AbLang-2 (45M) [Olsen et al., 2024], AMPLIFY (120M and 350M) [Fournier et al., 2024], DiffAbOpt [Raghu et al., 2025] (a CDR-only model that also conditions on a seed's predicted structure [Passaro et al., 2025]), and an internally-trained 13M parameter BERT-style model [Devlin et al., 2019] trained on [Dunbar et al., 2014]; all but ESM-2 were trained on [OAS; Kovaltsuk et al., 2018] antibody sequences. We also evaluated three CLMs: pIgGen and pIgGen-developability (17M) [Turnbull et al., 2024], and CloneLM (377M) [Amin et al., 2024], all of which were trained on antibodies. These methods were evaluated on a real scFv antibody therapeutics program for which we have supervised oracles for predicting synthesizability, binding affinity, and thermostability (melting temperature); we also evaluated OASis percentile and isoelectric point. Each method was given the same set of 10 seeds; we evaluated generated variant sequences with exactly 3 substitutions.

An initial evaluation of sampling algorithms on AbLang2 found that Gibbs sampling outperformed denoising in predicted synthesizability, and produced more diversity than Gibbs-argmax (Appendix Figure 11). We therefore focused our evaluations of MLMs with these samplers, and of CLMs with our beam search.

Overall, the AbLang2 and ESM2-650M models distinguished themselves with excellent metrics across a variety of criteria, and our beam search outperformed Gibbs sampling. We first show results for predicted synthesizability (Fig. 2A) and predicted thermostability (Fig 2B). Methods using our proposed beam search method consistently performed better on predicted synthesizability, compared to other sampling methods; in contrast, there was no discernable improvement for thermostability.

Pairwise diversity results are shown in Fig. 2C for inter-seed diversity and Fig 2D for intra-seed diversity. We see that beam search has much less intra-seed diversity than Gibbs, but this pattern is much less strong for inter-seed diversity. More generally, we see that intra-seed and inter-seed diversity are not necessarily correlated; CloneLM Beam and Amplify Beam have decent intra-seed diversity but low inter-seed diversity. Low intra-seed diversity means that children of the same seed tend to share the same mutated positions and especially the same new residues, whereas low inter-seed diversity means that variants from different seeds are converging to the same part of sequence space; the latter is more problematic.

See Appendix A.8 for further experimental details and results (on humanness, germline bias, isoelectric point, and CDR mutation count).

3.3 *In vitro* benchmarking

When choosing a smaller set of models for *in vitro* evaluation, we brought forward AbLang2 and ESM2-650M for their *in silico* performance. We also evaluated Sapiens as a small-parameter-count baseline model, pIgGen-dev as a representative of CLMs, SAbDabMLM due to its non-OAS training data, and DiffAb+ to represent structure-based methods. We applied these models to a fresh FAb antibody therapeutic campaign, evaluating chiefly on their ability to pass synthesizability and binding QC criteria. We used an initial candidate hit as seed; for each method 2/3 of the sequences sent to the wet lab had exactly 4 CDR-only substitutions, and 1/3 had exactly 3 substitutions in any location. (Since DiffAb+ is a CDR-only method, all its variants had exactly 4 CDR-only substitutions.)

Later, after obtaining 729 samples and training a supervised classification model for synthesizability and binding success, we evaluated the effect of combining this with AbLang2 when proposing batches. We use the model for post-MLM filtering ($p_{success} > 0.6$), post-MLM ranking, and finally NDS and STS guidance with post-MLM ranking.

Results, comprising 289 samples across 14 methods, each with at least 21 samples, are shown in Figure 3. The overall success rate of each method is shown in Fig 3A. Consistent with *in silico* results, ESM2-650M and AbLang2 had the best unsupervised success rates. Also, our proposed beam search outperformed Gibbs on all three models where both were tried. On Sapiens, Gibbs-argmax outperformed Gibbs, but it struggled to propose many unique new sequences. We found that using supervision to rank the unguided AbLang2 outputs improved success rate considerably. Using NDS and STS MOO guidance during generation improved results even further, with STS MOO guidance leading to a perfect 100% success rate.

In Fig 3B, we see that methods with higher success rate usually had tighter binding on their successful antibodies. Notable exceptions were SAbDabMLM and pIgGen-dev, perhaps due to their idiosyncratic training data. Both NDS and STS guidance not only improved success rate, but eliminated generation of very weak binders. In Fig 3C, for yield relative to CFPS input volume (a quantitative measure of synthesizability), we do not see any substantial trends among unguided methods. We do see, though, that our guidance methods produced antibodies with less variance in yield, particularly on the higher end of the distribution.

We show additional *in silico* evaluations of the *in vitro* experiment sequences in Fig 4 and Fig 5. In Fig 4A, showing OASis humanness, we once again witness a pattern of superiority for our proposed beam search methods over their Gibbs counterparts. On the other hand, we surprisingly see that AbLang2 tended to produce less human antibodies, despite being trained on human sequences; meanwhile, ESM2 650M had fairly high humanness outputs, despite not even being trained on antibody-specific data. SAbDabMLM sequences have low humanness, unsurprisingly since it was not trained on OAS. We do see that the guidance methods produced less human antibodies, a risk of using a single guidance model which can in principle be ameliorated by guidance with a humanness metric.

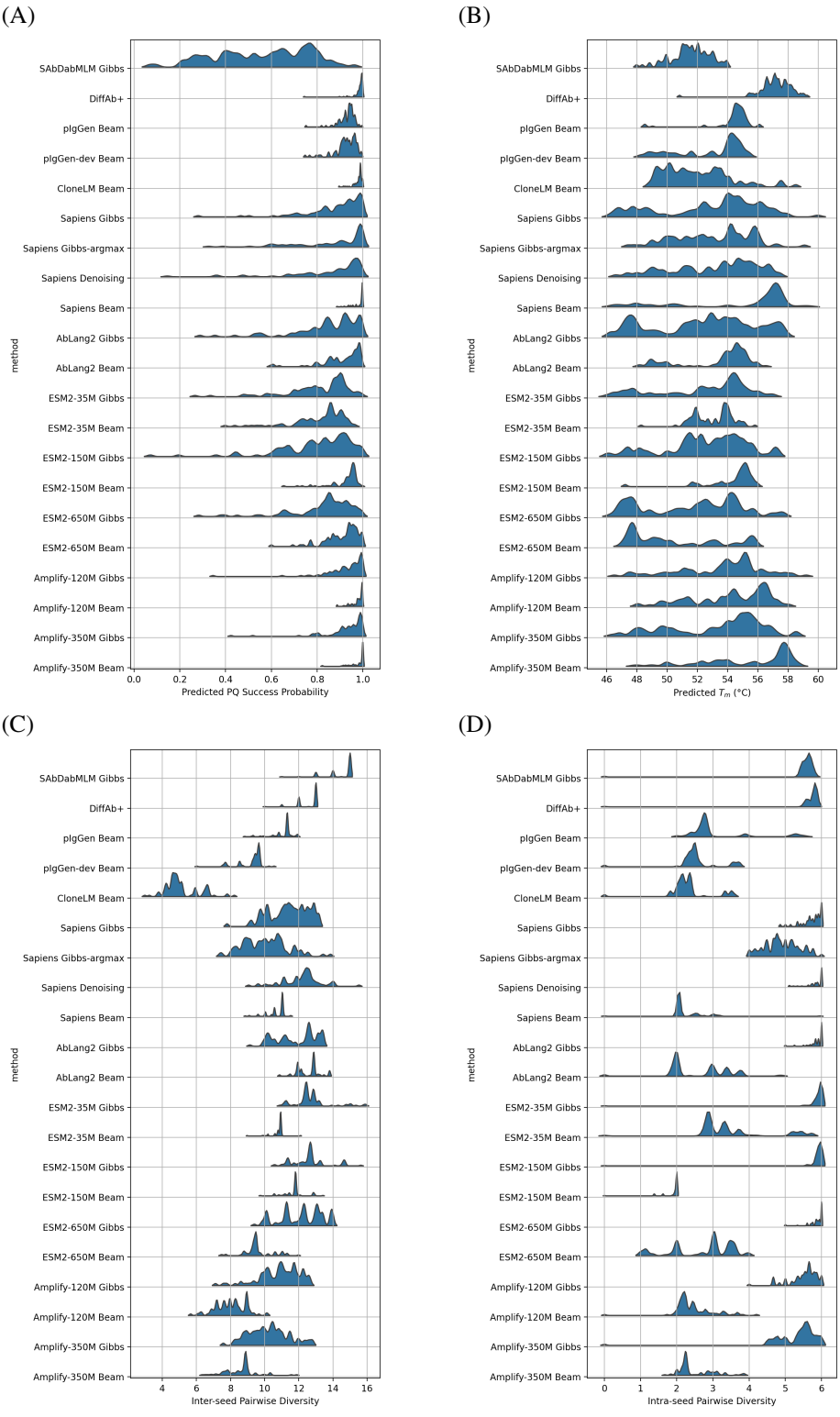


Figure 2: *In silico* metrics of quality and diversity for generated sequences: (A) predicted probability of synthesizability, (B) predicted melting temperature, (C) inter-seed pairwise diversity, and (D) intra-seed pairwise diversity.

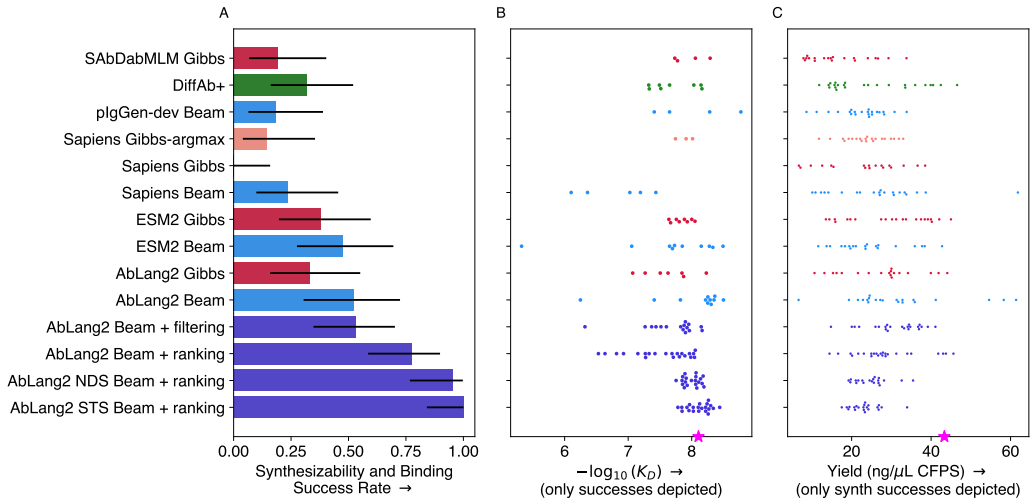


Figure 3: *In vitro* results. For error bars around the success rate, we show 95% CIs via binomial test inversion. We depict the seed's binding affinity and yield as a star on the x-axis.

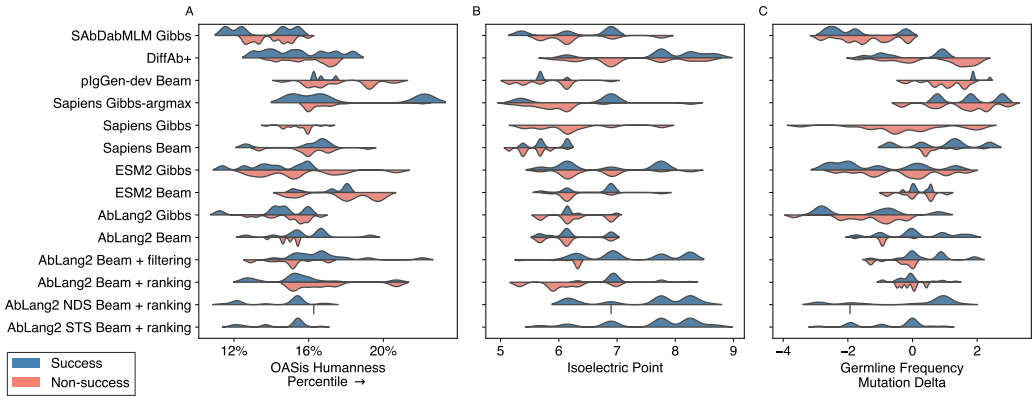


Figure 4: Additional developability-related *in silico* evaluations of the sequences in the *in vitro* experiment. For each method, observations are split by successful synthesizability and binding.

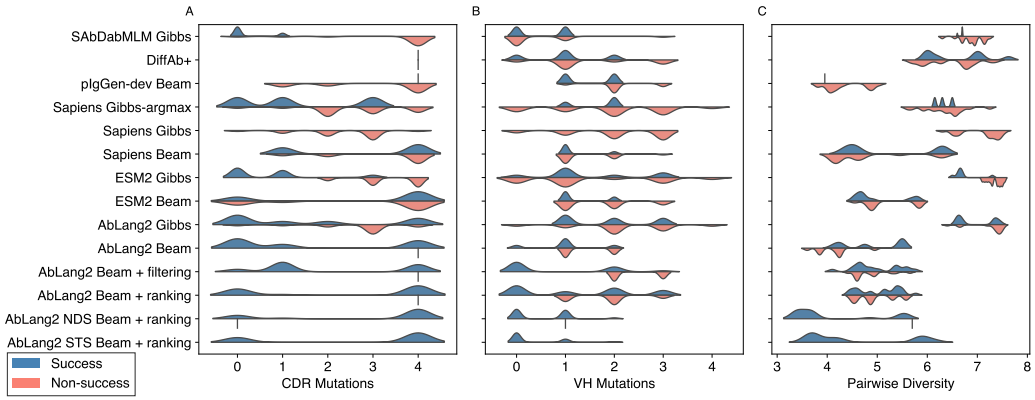


Figure 5: Additional analysis of mutational preferences of different methods for the *in vitro* experiment. or each method, observations are split by successful synthesizability and binding.

In Fig 4C, we show germline bias. Positive values indicate mutations toward more germline-typical residues. Recall that while increased humanness is usually good for antibody therapeutics optimization (i.e. reduced immunogenicity), positive germline bias can be problematic for binding specificity [Olsen et al., 2024]. We see that Sapiens Gibbs-argmax and Sapiens Beam, but not Sapiens Gibbs, have high germline bias. This is consistent with the observation in Olsen et al. [2024] that Sapiens has high germline bias, and with our observation that Gibbs sampling often fails to elicit sequences that are consistent with that model's preferences.

In Fig 5A, we reassuringly see that the successful supervised methods still heavily mutated the CDR regions, like the unsupervised methods. On the other hand, in Fig 5B, we see that they were systematically different from unsupervised methods in that they more frequently mutated the VL rather than the VH chain. In Fig 5C, showing pairwise diversity, we see that our beam search method and especially our guidance-based methods had less pairwise diversity. However, it's important to note that this experiment reflects diversity when using a single seed. As shown in our *in silico* evaluation, beam search had less intra-seed diversity, while maintaining inter-seed diversity.

4 Conclusion

We proposed a new MLM sampling method for iterative protein optimization, departing from the dominant mutation-centric paradigm and replacing it with sequence-centric evaluation and search. We benchmarked sampling methods and several language models on real antibody therapeutics optimization campaigns. Our results yield new insights into the effects of models, sampling algorithms, and guidance.

We conclude with several practical recommendations. First, we advocate for the use of supervision, for both ranking and guidance, whenever labeled data may be acquired. Second, we are able to recommend ESM2-650M and especially AbLang2 for antibody engineering. Third, we advise using our proposed stochastic beam search over Gibbs-type sampling. Fourth, we highlight the need for care when using supervised guidance, as it can lead to undesirable side effects in multi-objective optimization settings. Fifth, we suggest examining smooth Tchebycheff scalarization as an alternative to non-dominated Pareto-based sorting, for settings where one wants to simultaneously maximally-satisfy multiple objectives, rather than make progress along each objective.

References

David H Ackley, Geoffrey E Hinton, and Terrence J Sejnowski. A learning algorithm for boltzmann machines. *Cognitive science*, 9(1):147–169, 1985.

Sarah Alamdari, Nitya Thakkar, Rianne Van Den Berg, Neil Tenenholtz, Robert Strome, Alan M Moses, Alex X Lu, Nicolò Fusi, Ava P Amini, and Kevin K Yang. Protein generation with evolutionary diffusion: sequence is all you need. *BioRxiv*, pages 2023–09, 2023.

Alan Nawzad Amin, Nate Gruver, Yilun Kuang, Lily Li, Hunter Elliott, Calvin McCarter, Aniruddh Raghu, Peyton Greenside, and Andrew Gordon Wilson. Bayesian optimization of antibodies informed by a generative model of evolving sequences. *arXiv preprint arXiv:2412.07763*, 2024.

Nathaniel R Bennett, Joseph L Watson, Robert J Ragotte, Andrew J Borst, Déjenaé L See, Connor Weidle, Riti Biswas, Ellen L Shrock, Philip JY Leung, Buwei Huang, et al. Atomically accurate de novo design of single-domain antibodies. *bioRxiv: The Preprint Server for Biology*, 2024.

Aadyot Bhatnagar, Sarthak Jain, Joel Beazer, Samuel C Curran, Alexander M Hoffnagle, Kyle S Ching, Michael Martyn, Stephen Nayfach, Jeffrey A Ruffolo, and Ali Madani. Scaling unlocks broader generation and deeper functional understanding of proteins. *bioRxiv*, pages 2025–04, 2025.

Andrew Buchanan, Eric Bennett, Rebecca Croasdale-Wood, Andreas Evers, Brian Fennell, Norbert Furtmann, Konrad Krawczyk, Sandeep Kumar, Christopher James Langmead, Melody Shahsavarian, et al. How to think about designing smart antibodies in the age of genai: integrating biology, technology, and experience. In *MAbs*, volume 17, page 2490790. Taylor & Francis, 2025.

Sarah M Burbach and Bryan Briney. A curriculum learning approach to training antibody language models. *PLOS Computational Biology*, 21(9):e1013473, 2025.

Paul J Carter and Greg A Lazar. Next generation antibody drugs: pursuit of the'high-hanging fruit'. *Nature Reviews Drug Discovery*, 17(3):197–223, 2018.

Tong Chen, Yinuo Zhang, Sophia Tang, and Pranam Chatterjee. Multi-objective-guided discrete flow matching for controllable biological sequence design. *arXiv preprint arXiv:2505.07086*, 2025.

Lewis Chinery, Alissa M Hummer, Brij Bhushan Mehta, Rahmad Akbar, Puneet Rawat, Andrei Slabodkin, Khang Le Quy, Fridtjof Lund-Johansen, Victor Greiff, Jeliazko R Jeliazkov, et al. Simple computational methods can outperform deep learning in designing diverse, binder-enriched antibody libraries. *bioRxiv*, pages 2024–03, 2024.

Cyrus Chothia and Arthur M Lesk. Canonical structures for the hypervariable regions of immunoglobulins. *Journal of molecular biology*, 196(4):901–917, 1987.

Peter JA Cock, Tiago Antao, Jeffrey T Chang, Brad A Chapman, Cymon J Cox, Andrew Dalke, Iddo Friedberg, Thomas Hamelryck, Frank Kauff, Bartek Wilczynski, et al. Biopython: freely available python tools for computational molecular biology and bioinformatics. *Bioinformatics*, 25(11):1422, 2009.

UniProt Consortium. Uniprot: a hub for protein information. *Nucleic acids research*, 43(D1): D204–D212, 2015.

Jeremie Theddy Darmawan, Yarin Gal, and Pascal Notin. Sampling protein language models for functional protein design. In *Learning Meaningful Representations of Life (LMRL) Workshop at ICLR 2025*, 2025.

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

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186, 2019.

Frédéric A Dreyer, Daniel Cutting, Constantin Schneider, Henry Kenlay, and Charlotte M Deane. Inverse folding for antibody sequence design using deep learning. *arXiv preprint arXiv:2310.19513*, 2023.

James Dunbar and Charlotte M Deane. Anarci: antigen receptor numbering and receptor classification. *Bioinformatics*, 32(2):298–300, 2016.

James Dunbar, Konrad Krawczyk, Jinwoo Leem, Terry Baker, Angelika Fuchs, Guy Georges, Jiye Shi, and Charlotte M Deane. Sabdab: the structural antibody database. *Nucleic acids research*, 42(D1):D1140–D1146, 2014.

Quentin Fournier, Robert M Vernon, Almer van der Sloot, Benjamin Schulz, Sarath Chandar, and Christopher James Langmead. Protein language models: Is scaling necessary? *bioRxiv*, pages 2024–09, 2024.

Stuart Geman and Donald Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on pattern analysis and machine intelligence*, pages 721–741, 1984.

Marjan Ghazvininejad, Omer Levy, Yinhan Liu, and Luke Zettlemoyer. Mask-predict: Parallel decoding of conditional masked language models. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pages 6112–6121, 2019.

Kevin Gimpel, Dhruv Batra, Chris Dyer, and Gregory Shakhnarovich. A systematic exploration of diversity in machine translation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1100–1111, 2013.

Cade Gordon, Aniruddh Raghu, Peyton Greenside, and Hunter Elliott. Generative humanization for therapeutic antibodies. *arXiv preprint arXiv:2412.04737*, 2024.

Emil Julius Gumbel. *Statistical theory of extreme values and some practical applications: a series of lectures*, volume 33. US Government Printing Office, 1954.

Thomas Hayes, Roshan Rao, Halil Akin, Nicholas J. Sofroniew, Deniz Oktay, Zeming Lin, Robert Verkuil, Vincent Q. Tran, Jonathan Deaton, Marius Wiggert, Rohil Badkundri, Irhum Shafkat, Jun Gong, Alexander Derry, Raul S. Molina, Neil Thomas, Yousuf A. Khan, Chetan Mishra, Carolyn Kim, Liam J. Bartie, Matthew Nemeth, Patrick D. Hsu, Tom Sercu, Salvatore Candido, and Alexander Rives. Simulating 500 million years of evolution with a language model. *Science*, 2025. doi: 10.1126/science.ads0018. URL `http://dx.doi.org/10.1126/science.ads0018`.

Brian L Hie, Varun R Shanker, Duo Xu, Theodora UJ Bruun, Payton A Weidenbacher, Shaogeng Tang, Wesley Wu, John E Pak, and Peter S Kim. Efficient evolution of human antibodies from general protein language models. *Nature biotechnology*, 42(2):275–283, 2024.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*, 2019.

Emiel Hoogeboom, Alexey A Gritsenko, Jasmijn Bastings, Ben Poole, Rianne van den Berg, and Tim Salimans. Autoregressive diffusion models. *arXiv preprint arXiv:2110.02037*, 2021.

Thomas A Hopf, Artem Gazizov, Sergio Garcia Busto, Ethan Eschbach, SunJae Lee, Milot Mirdita, Rose Orenbuch, Khaoula Belahsen, David Ross, Chris Sander, et al. evedesign: accessible biosequence design with a unified framework. *bioRxiv*, pages 2026–03, 2026.

Sean R Johnson, Sarah Monaco, Kenneth Massie, and Zaid Syed. Generating novel protein sequences using gibbs sampling of masked language models. *bioRxiv*, pages 2021–01, 2021.

Elvin A Kabat and Tai Te Wu. Attempts to locate complementarity-determining residues in the variable positions of light and heavy chains. *Annals of the New York Academy of Sciences*, 190 (1):382–393, 1971.

Nal Kalchbrenner, Lasse Espeholt, Karen Simonyan, Aaron van den Oord, Alex Graves, and Koray Kavukcuoglu. Neural machine translation in linear time. *arXiv preprint arXiv:1610.10099*, 2016.

Scott Kirkpatrick, C Daniel Gelatt Jr, and Mario P Vecchi. Optimization by simulated annealing. *science*, 220(4598):671–680, 1983.

Georges Köhler and Cesar Milstein. Continuous cultures of fused cells secreting antibody of predefined specificity. *nature*, 256(5517):495–497, 1975.

Wouter Kool, Herke Van Hoof, and Max Welling. Stochastic beams and where to find them: The gumbel-top-k trick for sampling sequences without replacement. In *International conference on machine learning*, pages 3499–3508. PMLR, 2019.

Aleksandr Kovaltsuk, Jinwoo Leem, Sebastian Kelm, James Snowden, Charlotte M Deane, and Konrad Krawczyk. Observed antibody space: a resource for data mining next-generation sequencing of antibody repertoires. *The Journal of Immunology*, 201(8):2502–2509, 2018.

Yun-Huai Kuo, Chloe N Brown, Ece Akin, Yunxuan Xie, and Peter M Tessier. Multi-objective antibody design and optimization using machine learning. *Nature Reviews Bioengineering*, pages 1–15, 2026.

Jason Lee, Elman Mansimov, and Kyunghyun Cho. Deterministic non-autoregressive neural sequence modeling by iterative refinement. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1173–1182, 2018.

Marie-Paule Lefranc, Christelle Pommié, Manuel Ruiz, Véronique Giudicelli, Elodie Foulquier, Lisa Truong, Valérie Thouvenin-Contet, and Gérard Lefranc. Imgt unique numbering for immunoglobulin and t cell receptor variable domains and ig superfamily v-like domains. *Developmental & Comparative Immunology*, 27(1):55–77, 2003.

Xiner Li, Yulai Zhao, Chenyu Wang, Gabriele Scalia, Gokcen Eraslan, Surag Nair, Tommaso Biancalani, Shuiwang Ji, Aviv Regev, Sergey Levine, et al. Derivative-free guidance in continuous and discrete diffusion models with soft value-based decoding. *arXiv preprint arXiv:2408.08252*, 2024.

Xi Lin, Xiaoyuan Zhang, Zhiyuan Yang, Fei Liu, Zhenkun Wang, and Qingfu Zhang. Smooth tchebycheff scalarization for multi-objective optimization. *arXiv preprint arXiv:2402.19078*, 2024.

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

Ruei-Min Lu, Yu-Chyi Hwang, I-Ju Liu, Chi-Chiu Lee, Han-Zen Tsai, Hsin-Jung Li, and Han-Chung Wu. Development of therapeutic antibodies for the treatment of diseases. *Journal of biomedical science*, 27(1):1, 2020.

Ali Madani, Bryan McCann, Nikhil Naik, Nitish Shirish Keskar, Namrata Anand, Raphael R Eguchi, Po-Ssu Huang, and Richard Socher. Progen: Language modeling for protein generation. *arXiv preprint arXiv:2004.03497*, 2020.

Chris J Maddison, Daniel Tarlow, and Tom Minka. A* sampling. *Advances in neural information processing systems*, 27, 2014.

Derek M Mason, Simon Friedensohn, Cédric R Weber, Christian Jordi, Bastian Wagner, Simon M Meng, Roy A Ehling, Lucia Bonati, Jan Dahinden, Pablo Gainza, et al. Optimization of therapeutic antibodies by predicting antigen specificity from antibody sequence via deep learning. *Nature biomedical engineering*, 5(6):600–612, 2021.

Frederick A Matsen IV, Will Dumm, Kevin Sung, Mackenzie M Johnson, David H Rich, Tyler N Starr, Yun S Song, Julia Fukuyama, and Hugh K Haddox. Separating selection from mutation in antibody language models. *Elife*, 15:RP109644, 2026.

John McCafferty, Andrew D Griffiths, Greg Winter, and David J Chiswell. Phage antibodies: filamentous phage displaying antibody variable domains. *nature*, 348(6301):552–554, 1990.

Joshua Meier, Roshan Rao, Robert Verkuil, Jason Liu, Tom Sercu, and Alex Rives. Language models enable zero-shot prediction of the effects of mutations on protein function. *Advances in neural information processing systems*, 34:29287–29303, 2021.

Clara Meister, Ryan Cotterell, and Tim Vieira. If beam search is the answer, what was the question? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2173–2185, 2020.

Hunter Nisonoff, Junhao Xiong, Stephan Allenspach, and Jennifer Listgarten. Unlocking guidance for discrete state-space diffusion and flow models. *arXiv preprint arXiv:2406.01572*, 2024.

Pascal Notin, Mafalda Dias, Jonathan Frazer, Javier Marchena-Hurtado, Aidan N Gomez, Debora Marks, and Yarin Gal. Tranception: protein fitness prediction with autoregressive transformers and inference-time retrieval. In *International Conference on Machine Learning*, pages 16990–17017. PMLR, 2022.

Pascal Notin, Aaron Kollasch, Daniel Ritter, Lood Van Niekerk, Steffanie Paul, Han Spinner, Nathan Rollins, Ada Shaw, Rose Orenbuch, Ruben Weitzman, et al. Proteingym: Large-scale benchmarks for protein fitness prediction and design. *Advances in neural information processing systems*, 36: 64331–64379, 2023.

Sebastian W Ober, Calvin McCarter, Aniruddh Raghu, Yucen Lily Li, Alan N Amin, Andrew Gordon Wilson, and Hunter Elliott. Is sequence information all you need for bayesian optimization of antibodies? *arXiv preprint arXiv:2509.24933*, 2025.

Tobias H Olsen, Iain H Moal, and Charlotte M Deane. Addressing the antibody germline bias and its effect on language models for improved antibody design. *Bioinformatics*, 40(11):btae618, 2024.

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.

David Prihoda, Jad Maamary, Andrew Waight, Veronica Juan, Laurence Fayadat-Dilman, Daniel Svozil, and Danny A Bitton. Biophi: A platform for antibody design, humanization, and humanness evaluation based on natural antibody repertoires and deep learning. In *MAbs*, volume 14, page 2020203. Taylor & Francis, 2022.

Aniruddh Raghu, Sebastian Ober, Maxwell Kazman, and Hunter Elliott. Guided sequence-structure generative modeling for iterative antibody optimization. *arXiv preprint arXiv:2509.16357*, 2025.

Jackie Rao, Ferran Gonzalez Hernandez, Leon Gerard, and Alexandra Gessner. Boat: Navigating the sea of in silico predictors for antibody design via multi-objective bayesian optimization. *arXiv preprint arXiv:2604.13980*, 2026.

Subham Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin Chiu, Alexander Rush, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. *Advances in Neural Information Processing Systems*, 37:130136–130184, 2024.

Julian Salazar, Davis Liang, Toan Q Nguyen, and Katrin Kirchhoff. Masked language model scoring. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 2699–2712, 2020.

Yair Schiff, Subham Sekhar Sahoo, Hao Phung, Guanghan Wang, Sam Boshar, Hugo Dalla-torre, Bernardo P de Almeida, Alexander Rush, Thomas Pierrot, and Volodymyr Kuleshov. Simple guidance mechanisms for discrete diffusion models. *arXiv preprint arXiv:2412.10193*, 2024.

Richard W Shuai, Jeffrey A Ruffolo, and Jeffrey J Gray. Generative language modeling for antibody design. *BioRxiv*, pages 2021–12, 2021.

Sophia Tang, Yinuo Zhang, and Pranam Chatterjee. Peptune: De novo generation of therapeutic peptides with multi-objective-guided discrete diffusion. *ArXiv*, pages arXiv–2412, 2025.

Oliver M Turnbull, Dino Oglic, Rebecca Croasdale-Wood, and Charlotte M Deane. p-iggen: a paired antibody generative language model. *Bioinformatics*, 40(11):btae659, 2024.

Tim Vieira. Gumbel-max trick and weighted reservoir sampling, 2014. URL `http://timvieira.github.io/blog/gumbel-max-trick-and-weighted-reservoir-sampling/`.

Alex Wang and Kyunghyun Cho. Bert has a mouth, and it must speak: Bert as a markov random field language model. *arXiv preprint arXiv:1902.04094*, 2019.

Kevin K Yang, Nicolo Fusi, and Alex X Lu. Convolutions are competitive with transformers for protein sequence pretraining. *Cell systems*, 15(3):286–294, 2024.

Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015.

Ruoxi Zhang and Pranam Chatterjee. Multi-objective nanobody design via masked discrete diffusion with simplex refinement. In *NeurIPS 2025 Workshop on Structured Probabilistic Inference $\{\backslash\&\}$ Generative Modeling*, 2025.

A Appendix

A.1 An empirical evaluation of the masked-marginal approximation

We performed a deeper dive into the validity of our approximation for a larger set of seed sequences, for paired-chain vs unpaired-chain modeling, and for CDR-region-only vs CDR+framework full-sequence. The results, reinforce the validity of the approximation, and are also interesting in their own right. In summary, we extended the empirical analysis from our earlier rebuttal, and almost always observed the same strong correlation between approximate and true PLLs.

The exception is an insightful corner case: it struggles only near-germline seeds, intersected with framework mutations, intersected with AbLang2. Digging deeper, we found strong evidence that this is aberrant behavior with AbLang2, where its "true" PLL landscape has non-biological anomalies from memorizing germline framework sequences, as opposed to true biological epistasis. (One would avoid using an LM with such behavior on such regions and seeds, so the weak correlation is a moot point. And our approximate-PLL was indeed better biologically than the "true-PLL" when they disagreed.)

Our main findings, on the 7 antibody panel from Hie et al. [2024], showing the approximate-vs-true PLL Spearman correlations, on full (framework and CDR) VH-plus-VL sequences, are shown in Table 2. We see strong approximate-vs-true Spearman correlations for Sapiens (antibody LM) and ESM2-650M, and even for AbLang2 except for antibodies `medi8852-uca` and `mab114-uca`. Note the "-uca" (unmutated common ancestor, UCA): these are unobserved but inferred consensus germline sequences. Because UCAs are inferred germline ancestors with no experimental structures, inverse-folding MPNNs were skipped.

Antibody	Sapiens	AbLang2	ESM2-650M
`c143`	0.941	0.955	0.930
`mab114`	0.924	0.935	0.896
`mab114-uca`	0.946	0.734	0.928
`medi8852`	0.837	0.925	0.878
`medi8852-uca`	0.878	0.282	0.888
`regn10987`	0.910	0.933	0.942
`s309`	0.879	0.854	0.929

Table 2: Approximate-vs-true PLL Spearman correlation, one-edit full VH+VL variants, over the seven-antibody panel from Hie et al. [2024].

We will now provide evidence that AbLang2's "true" PLL landscape is degenerate on UCA seeds due to germline framework memorization. First, we isolate the weak correlation on these seeds to the frameworks. Second, we show that AbLang2 has extremely low perplexity on these seeds, surrounded by landscape cliffs, indicating memorization. Finally, we show that DASM [Matsen IV et al., 2026], a new model that explicitly controls for "germline bias" in a principled manner, has better correlation with the approximate PLLs than the true PLLs, precisely on these seeds with AbLang2. In other words, near the germline where the approximate-AbLang2 and true-AbLang2 disagree, the true-AbLang2 is the *odd one out* compared to approximate-AbLang2, DASM, and the other MLMs' PLLs.

In this paragraph, we provide some additional background on antibody "germline bias", which may be useful to some readers for the subsequent analysis. The human body does not store a catalog of finished antibodies, but rather a few hundred inherited gene fragments (the "germline" segments), which get shuffled and then locally mutated to produce each mature antibody. Most of the sequence (the "framework") stays close to the inherited template, while the mutations concentrate in three sequence regions (the CDRs) that actually touch the antigen that is being targeted by the immune system. Because every LM training sequence is therefore a lightly-edited copy of one of those few hundred templates, an antibody LM can score well on masked-token recovery by filling in the germline amino acids. But such a model then proposes germline reversions, assigning low likelihood to exactly the mutations that confer binding to specific antigens. AbLang2 was the first LM that attempted to avoid germline bias, via ML-based heuristics. We hypothesize that AbLang2 avoided having its entire likelihood landscape sloped towards germlines, by instead concentrating probability

mass onto the immediate neighborhood of germline sequences (thus making them extremely high germline bias), while being relatively flat and low germline bias below these "cliffs". DASM is a newer model that seeks to directly model the above process to eliminate germline bias, though it is not an LM, providing a relative "selection factor" given a (sequence, mutation)-tuple, rather than a PLL given a sequence.

Now we provide multiple independent pieces of evidence that the "true" AbLang2 PLLs on these UCAs are erroneous and degenerate.

First, we isolate that the disagreement between "true" and masked-marginals comes from the inclusion of framework region mutations. We exclude framework mutations, including only CDR region (Chothia definition) positions into the PLL; we still condition on the frameworks by including the full sequence in the forward pass. As shown in Table 3, this mostly rescues the UCA correlations on AbLang2, while leaving others basically untouched.

Antibody	Sapiens-CDR	AbLang2-CDR	ESM2-CDR
c143	0.971	0.930	0.892
mab114	0.933	0.859	0.880
mab114-uca	0.946	0.855	0.946
medi8852	0.828	0.881	0.906
medi8852-uca	0.778	0.675	0.895
regn10987	0.887	0.927	0.919
s309	0.898	0.831	0.917

Table 3: Approximate-vs-true PLL Spearman correlation over the seven-antibody panel from Hie et al., scoring only Chothia-defined CDR positions of the one-edit variant (while still conditioning on the full VH+VL sequence in the forward pass).

Second, we show that AbLang2 uniquely gives extremely large likelihoods specifically to the UCA seeds, revealing memorization. In Table 4, we show per-residue pseudo-perplexity of the full seed sequences. The UCA seeds have unusually low perplexity, particularly for the antibody LMs.

Antibody	Sapiens	AbLang2	ESM2
c143	1.878	1.955	3.301
mab114	2.107	1.949	3.967
mab114-uca	1.256	1.256	2.862
medi8852	1.607	1.761	3.719
medi8852-uca	1.212	1.156	3.653
regn10987	1.407	1.447	2.838
s309	1.559	1.640	2.834

Table 4: Per-residue pseudo-perplexity of the full seed sequence, over the seven-antibody panel from Hie et al. AbLang2 uniquely assigns extremely low perplexity to the UCA seeds, revealing memorization.

Third, we observe that AbLang2's UCA likelihood peaks are indeed surrounded by sharp cliffs in the local landscape. Table 5 shows the geometric mean pseudoperplexity ratios for the seeds, confirming the cliffs around the AbLang2 UCAs.

Finally, and most convincingly, we compare the PLLs against DASM's germline bias-free selection factors. We compute DASM selection factors for the seed and each variant's mutation; then we measure the Spearmans between approximate PLLs vs DASM, and between true vs DASM. In Table 6, on the exact (UCA, AbLang2) intersection where approximate-vs-true correlation is weak, we see that DASM agrees more with "approximate" rather than "true". Indeed, on these seeds, DASM is negatively correlated with the AbLang2 "true" PPLs! This confirms that the AbLang2 true PLL landscapes around UCAs reflect degeneracy driven by germline bias, not biological selection lost by the masked-marginal approximation.

Antibody	Sapiens	AbLang2	ESM2
c143	88.4	1123.0	124.0
mab114	70.6	1141.9	74.8
mab114-uca	175.8	6837.3	175.1
medi8852	119.3	1420.3	87.0
medi8852-uca	173.3	8668.2	94.4
regn10987	150.1	3189.8	191.5
s309	119.5	1826.4	238.2

Table 5: Geometric-mean pseudo-perplexity ratio of the seed's local one-edit neighborhood (neighbor residue vs. seed residue), over the seven-antibody panel from Hie et al. AbLang2's UCA likelihood peaks are surrounded by sharp cliffs in the local landscape.

	Sapiens		AbLang2		ESM2	
Antibody	Approx	True	Approx	True	Approx	True
c143	0.34	0.35	0.53	0.49	0.53	0.52
mab114	0.39	0.39	0.52	0.43	0.55	0.53
mab114-uca	0.08	0.08	0.33	-0.09	0.57	0.55
medi8852	0.24	0.19	0.45	0.30	0.52	0.51
medi8852-uca	0.11	-0.01	0.28	-0.35	0.54	0.52
regn10987	0.23	0.23	0.39	0.22	0.52	0.51
s309	0.25	0.27	0.44	0.16	0.54	0.54

Table 6: Spearman correlation of each model's approximate ($\approx$) and true PLL against the DASM germline-bias-free selection score, per antibody, over the seven-antibody panel from Hie et al. [2024]. On the UCA seeds where AbLang2's approximate-vs-true correlation is weakest, DASM agrees more with the approximation than with the true PLL – and is negatively correlated with the AbLang2 true PLL, indicating the "true" landscape reflects germline bias rather than biological selection.

A.2 Approximations to the pseudo-log-likelihood in beam search

Recall that our beam search scores each candidate sequence $\boldsymbol{x}'$ (which differs from the template $\boldsymbol{x}$ by a substitution at position k) using an approximate PLL (Eq. 3). Here we discuss the space of approximations and their computational tradeoffs.

The exact PLL of $\boldsymbol{x}'$ is

$$PLL(\boldsymbol{x}') = \sum_{i=1}^{L} \log \hat{p}(x'_i \mid \boldsymbol{x}'_{j\neq i}), \tag{4}$$

where each term requires masking position i in the *child* sequence $\boldsymbol{x}'$ and running a forward pass. Since $\boldsymbol{x}'$ differs from $\boldsymbol{x}$ at position k, the conditional distributions $\hat{p}(\cdot \mid \boldsymbol{x}'_{j\neq i})$ differ from those of the template $\hat{p}(\cdot \mid \boldsymbol{x}_{j\neq i})$ at every position $i \neq k$ (because the context includes x'_k instead of x_k). Computing the exact PLL therefore requires L forward passes *per child*. With $O(L)$ children per beam search step (one per position per amino acid), the total cost is $O(L^4)$.

The key insight behind our beam search is that having computed the template's PLL – which requires L forward passes yielding the matrix $\boldsymbol{A}$ where $\boldsymbol{A}_{i,:} = \texttt{softmax}_\tau(\tilde{\boldsymbol{Y}}^{(i)}_{i,:})$ – one can cheaply approximate the PLL of all single-substitution neighbors. The question is what approximation to use at positions $i \neq k$.

Table 7 summarizes the five approaches. They are distinguished by what the model observes at positions i and k when computing the conditional probability at position $i \neq k$, as well as their computational cost. We note that position k always uses the exact single-mask conditional $\boldsymbol{A}_{k,\texttt{index}(x'_k)}$ in all approaches except the no-masking variants.

Table 7: Comparison of PLL approximations for scoring all single-substitution neighbors during one step of beam search. Position k is the substitution site; position $i \neq k$ is any other position. Each forward pass has cost $O(L^2)$.

Approximation	**Pos. i sees at i**	**Pos. i sees at k**	**Total cost**
Exact PLL	`MASK`	x'_k (child)	$O(L^4)$
Double-mask	`MASK`	`MASK`	$O(L^4)$
Masked-marginal	`MASK`	x_k (template)	$O(L^3)$
No-masking (child)	x'_i (visible)	x'_k (child)	$O(L^3)$
No-masking (template)	x_i (visible)	x_k (template)	$O(L^2)$

Masked-marginal (our approach). At position $i \neq k$, we use the template's own conditional distribution, i.e., the distribution obtained by masking position i in $\boldsymbol{x}$:

$$PLL_{\text{wt}}(\boldsymbol{x}'; \tau) = \sum_{i=1}^{L} \log \boldsymbol{A}_{i,\texttt{index}(x'_i)}. \tag{5}$$

Since $x'_i = x_i$ for $i \neq k$, only the term at $i = k$ differs from the template's PLL, so all neighbor PLLs can be read off from $\boldsymbol{A}$ with no additional forward passes. The approximation error is that position i is scored conditioned on seeing the *template* token x_k at position k, rather than the child's substituted token x'_k. The total cost is $O(L^3)$ (for the L stepwise-masked forward passes to compute $\boldsymbol{A}$), amortized over all $O(L)$ neighbors.

Double-mask. To avoid conditioning on the wrong token at position k, one can instead mask *both* positions i and k simultaneously. Let $\tilde{\boldsymbol{Y}}^{(i,k)}$ denote the MLM output when both positions i and k are masked. The approximate PLL becomes

$$PLL_{\text{dm}}(\boldsymbol{x}'; \tau) = \log \boldsymbol{A}_{k,\texttt{index}(x'k)} + \sum_{i \neq k} \log \texttt{softmax}_\tau(\tilde{\boldsymbol{Y}}^{(i,k)}_{i,\texttt{index}(x_i)}). \tag{6}$$

At position k we use the same single-mask conditional as the masked-marginal; at positions $i \neq k$, we are agnostic about the substitution site. Computing all pairwise double-mask outputs requires $\binom{L}{2}$ forward passes. Together with L single-mask forward passes, the total cost is $O(L^4)$. This is as expensive as the exact PLL, while introducing approximation error.

No-masking. At the other extreme, one can skip masking at position i entirely and use $\hat{p}(x_i \mid \boldsymbol{x})$ as an approximation to $\hat{p}(x_i \mid \boldsymbol{x}_{j \neq i})$ [Prihoda et al., 2022]. This can be applied to either the template sequence (one forward pass shared by all neighbors, $O(L^2)$ total) or each child sequence individually ($O(L^3)$ total, since each child requires its own forward pass). In either case, the model observes x_i itself when predicting position i, yielding probabilities near 1 that contribute little discriminative signal. At position k, the child no-masking variant at least conditions on the correct token x'_k (just without masking it), while the template no-masking variant conditions on the wrong token x_k without masking.

Discussion. The approaches are ordered in Table 7 from most to least principled. The exact PLL and the double-mask approach have the same asymptotic cost but differ in that the exact PLL conditions each position on the child's actual token at k, while the double-mask marginalizes over k via the mask token. In practice, the masked-marginal offers an appealing balance: it is the cheapest approach that still masks position i (yielding meaningful conditional probabilities), while the error from conditioning on the template token at k is empirically small in practice [Meier et al., 2021].

A.3 Multi-objective optimization guidance

For multi-objective optimization, we begin by converting the pseudo-log-likelihood scores to pseudo-perplexity scores. (Each sequence's pseudo-log-likelihood was already perturbed with independent Gumbel noise $g \sim \text{Gumbel}(0, 1)$ prior to computing the pseudo-perplexity.) Then, for each objective, we standardize scores to z-scores using the mean and standard deviation computed across

all candidate neighbors of all template sequences at the current number of edits. To produce a single ranking, we aggregate the z-scored objectives using a smooth Tchebycheff scalarization [Lin et al., 2024], computing $\log \sum_i \exp(c_i \tilde{f}_i)$ for each sequence, where $\tilde{f}_i$ are the z-scored objectives and c_i are per-objective weights, with larger-magnitude c_i placing greater emphasis on objective i; we use $c_i < 0$ for criteria which we seek to minimize. When computing the z-scores, we added $\epsilon = 0.01$ to the denominator for numerical stability. Sequences are then ranked by this score, with lower values preferred.

A.4 Additional experimental details for Trastuzumab optimization

Trastuzumab The seed is Trastuzumab's variable heavy domain (120 residues; UniProt sequence). We restrict mutations to the CDR-H3 segment defined by Mason et al. [2021] (the substring WGGDGFYAMD at residues 98-107, zero-indexed inclusive, of the VH); the framework regions, the VL, and the rest of the VH are held fixed.

Beam search At each step, the 20 children whose realized edit count strictly exceeds their parent's are propagated as the next frontier (so at depth k, every frontier sequence has exactly k realized edits). Each step also adds the top 200 children to a per-realized-edit-count output pool. The final return is the top 200 per realized-edit bucket. We use softmax temperature 1.5 throughout. The realized edit count tracked on each beam node is the Hamming distance to the seed; since we do not allow indels, no alignment is needed.

Mutation-centric baselines Gibbs sampling runs for E iterations per chain, sampling one mutable position uniformly at random, masking it, and resampling from the resulting MLM conditional at temperature $\tau = 1$; we run 400 chains per (model, E). Denoising sampling first picks E mutable positions uniformly without replacement, masks all of them simultaneously, and then unmasks one at a time over E iterations; at each iteration we select the masked position whose conditional distribution has the lowest entropy (the ESM-3 default; max-probability is supported but not used here), sample a residue from the conditional at that position, and commit. We run 400 chains per (model, E).

Guided variants Product-of-experts-guided Gibbs adds the supervised oracle's contribution to the MLM masked conditional before sampling. For the binary affinity CNN, the contribution at position k is $\log p_{\text{binder}}(\bar{x}; x_k = r)$ for each residue r, where $\bar{x}$ is the current sequence with position k replaced by r; we use guidance strength $\gamma = 1$. Beam-search STS guidance combines the MLM pseudo-perplexity (with weight $c_{\text{PLL}} = 1$) and one or two supervised objectives — the Mason CNN's predicted binder probability and (for the +OASis variants) the OASis percentile — each with weight $c = 2$ in the smooth-Tchebycheff scalarization (matching the recipe used for our in-vitro experiment).

Affinity oracles Both CNNs share the architecture of Mason et al. [2021]: `Conv1D(filters=400, kernel=5, padding=same, ReLU) → Dropout(0.2) → MaxPool1D(pool=2, stride=1) → Flatten → Dense(300, ReLU) → Dense(1, sigmoid)`. Inputs are 10-residue H3 substrings one-hot-encoded into a (10, 20) tensor in alphabetical AA order. We train with binary cross-entropy, Adam (lr 7.5e-5, batch size 16), and early stopping on validation loss with patience 5. We train two: (i) a strong evaluation oracle on the 367k training samples `her2-aff-large` [Chinery et al., 2024] random split (test ROC AUC 0.995, PR AUC 0.990; used to post-hoc score every generated sequence), and (ii) the Mason CNN guidance oracle on the deduplicated 34k-sample Mason CDR-H3 dataset (test ROC AUC 0.914, PR AUC 0.816 — matching the original reference). 2,342 sequences with conflicting binder/non-binder labels across the original Mason files were dropped before splitting.

A.5 Additional results for Trastuzumab optimization

AbLang2 results, presented in Appendix Figure 6, are qualitatively similar with one important twist: the AbLang2 PLL pulls strongly toward non-binder residues, so our unguided beam search performs poorly, like the mutation-centric baselines. In this scenario, `MasonCNN` guidance improves Gibbs PoE to `p(binder)` > 0.5 at $E = 6$, while `MasonCNN` guidance improves Beam STS to `p(binder)` > 0.9.

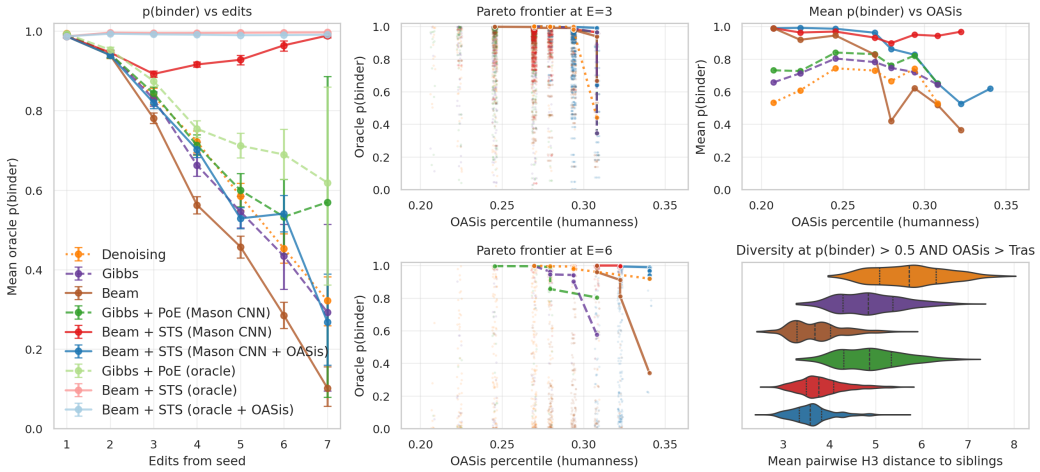


Figure 6: AbLang2 on the Trastuzumab CDR-H3 *in silico* evaluation. Color groups `MasonCNN`-guided methods with their lighter-shaded strong-oracle counterparts; linestyle indicates methodology (solid: ours; dashed: previous). **[Left:]** Mean predicted `p(binder)` vs edits from the seed; error bars depict 95%-bootstrap CIs. **[Middle:]** OASis × `p(binder)` Pareto frontiers. Faded dots show the underlying generated variants. **[Top Right:]** Mean `p(binder)` at each OASis percentile bucket, pooling across all edits. **[Bottom right:]** Pairwise H3-diversity violins restricted to "high-quality" variants — those with `p(binder)` > 0.5 AND OASis > 0.245, the Trastuzumab seed.

Table 8: Fraction of designs with oracle $p(\text{binder}) > 0.5$ at $E = 6$ edits, for the extended Trastuzumab models, both with (holo) and without (apo) the HER2 subdomain IV antigen.

Configuration	Beam	Gibbs	Beam:Mason	Gibbs:Mason	Beam:Mason+OASis
ESM2-650M	0.98	0.44	0.92	0.49	0.98
AbLang2	0.26	0.40	0.99	0.49	0.64
ProteinMPNN (holo)	0.39	0.41	0.72	0.42	0.94
ProteinMPNN (apo)	0.22	0.35	0.51	0.43	0.98
AbMPNN (holo)	0.73	0.54	0.92	0.61	0.84
AbMPNN (apo)	0.57	0.40	0.88	0.47	0.78

The results across all models are summarized in Table 8. We see that AbLang2 performs poorly; that ProteinMPNN and AbMPNN have performance strongly dependent on the antigen; that AbMPNN outperforms ProteinMPNN; and that ESM2-650M is at least as good as the inverse-folding MPNNs even with the antigen.

A.6 Hyperparameter sweep analysis results

As shown in Figure 9, the quality is quite flat as a function of these hyperparameters. Also, one can increase beam size to larger values than used in our experiments, thus increasing diversity, without substantially hurting the quality of generated variants.

A.7 Runtime analysis results

We measured the wall-clock seconds for the 7-edit Trastuzumab run on an A10G machine, with 1 process on the GPU. Results for models without guidance are shown in Figure 10. Note that this estimated advantage is smaller than the theoretical advantage, and dramatically underestimates the practical advantage because we designed this particular experiment, for fair quality evaluation, to generate and evaluate the same number of variants, only 200, from Gibbs and Beam. Without any extra cost, for Beam we can actually obtain all $lenMutableRegion \times (numAminoAcids - 1) \times beamSize = 3800$ scored and ranked variants instead (though with the caveats that the lower-

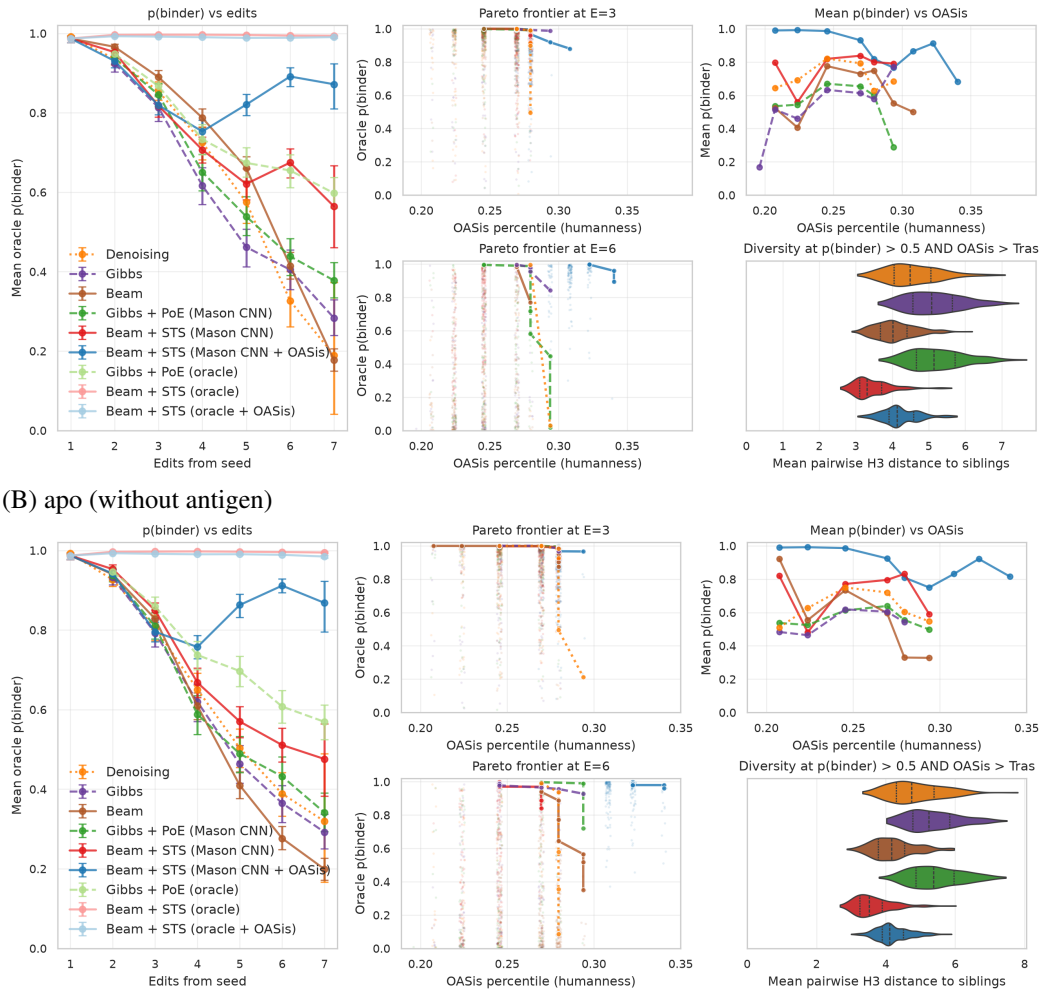


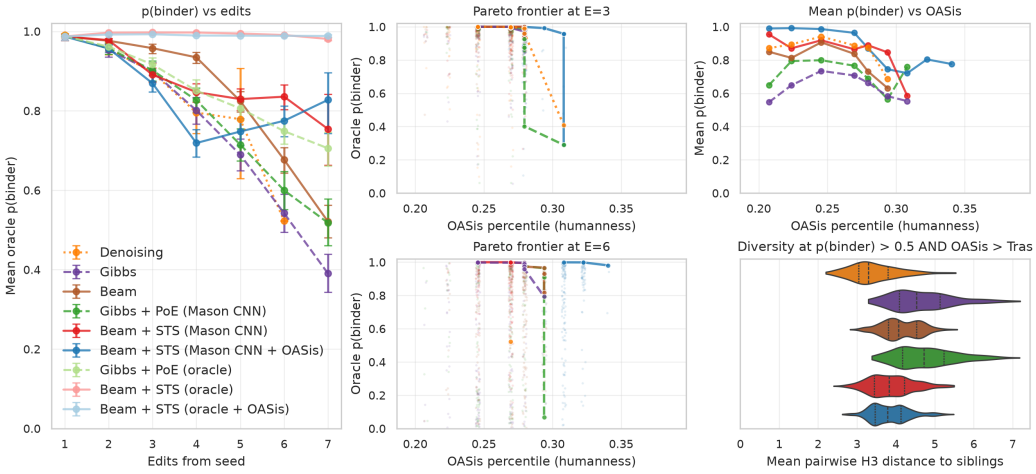
Figure 7: ProteinMPNN on the Trastuzumab CDR-H3 *in silico* evaluation, with (A) and without (B) the bound antigen. Color groups `MasonCNN`-guided methods with their lighter-shaded strong-oracle counterparts; linestyle indicates methodology (solid: ours; dashed: previous). **[Left:]** Mean predicted `p(binder)` vs edits from the seed; error bars depict 95%-bootstrap CIs. **[Middle:]** OASis × `p(binder)` Pareto frontiers. Faded dots show the underlying generated variants. **[Top Right:]** Mean `p(binder)` at each OASis percentile bucket, pooling across all edits. **[Bottom right:]** Pairwise H3-diversity violins restricted to "high-quality" variants — those with `p(binder)` > 0.5 AND OASis > 0.245, the Trastuzumab seed.

ranked variants will be of lower quality and that our scores use the approximate PLL); having all those variants available for downstream ranking and filtering is one of the advantages of our method.

A.8 Additional details for in silico benchmarking

Experimental setup We obtained predicted probabilities of synthesizability (i.e. protein quantification, PQ) from an oracle trained on 6038 samples, split 60%-20%-20% into train-valid-test sets. We trained a 5-model ensemble of 1d dilated convolutional neural networks [Yu and Koltun, 2015] with mean test AUROC 0.9057. We obtained melting temperature (T_m) predictions by finetuning ESM2-150M [Lin et al., 2022] on internal phage display and CFPS data. The model was trained to predict first melting temperature (in Celsius) measured by the Uncle (differential scanning fluorimetry) instrument's peak-calling algorithm on the Barycentric Mean (BCM) of intrinsic fluorescence. For samples from the lineage from which our seeds were drawn, the train-valid-test split was 384-

(A) holo (with antigen)



(B) apo (without antigen)

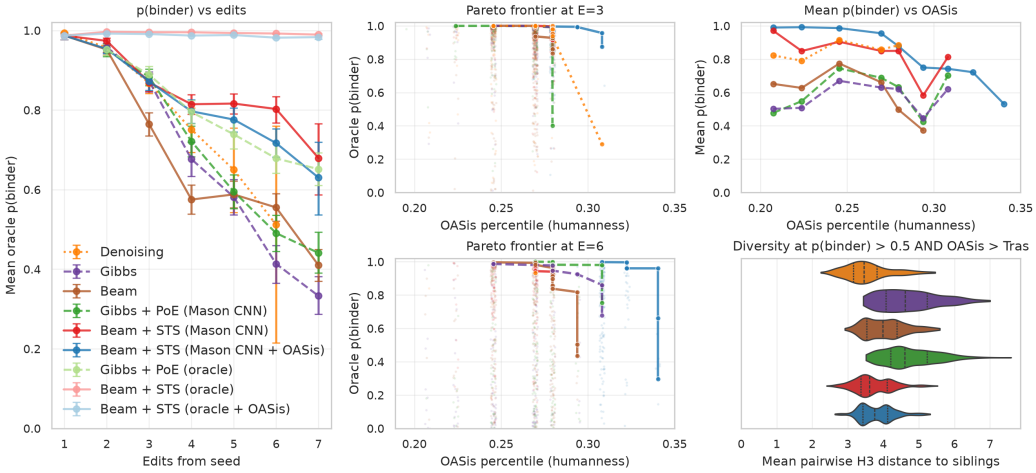


Figure 8: AbMPNN on the Trastuzumab CDR-H3 *in silico* evaluation, with (A) and without (B) the bound antigen. Color groups `MasonCNN`-guided methods with their lighter-shaded strong-oracle counterparts; linestyle indicates methodology (solid: ours; dashed: previous). **[Left:]** Mean predicted `p(binder)` vs edits from the seed; error bars depict 95%-bootstrap CIs. **[Middle:]** OASis × `p(binder)` Pareto frontiers. Faded dots show the underlying generated variants. **[Top Right:]** Mean `p(binder)` at each OASis percentile bucket, pooling across all edits. **[Bottom right:]** Pairwise H3-diversity violins restricted to "high-quality" variants — those with `p(binder)` > 0.5 AND OASis > 0.245, the Trastuzumab seed.

188-219 samples, with test Spearman correlation of 0.7693. On a larger test-set of 1111 samples from different lineages for the same therapeutics program, the test Spearman was 0.8486.

To compute intra-seed (inter-seed) pairwise diversity, for each child sequence, we align it to all other child sequences of the same (different) seed, and compute the number of mutations. For each child antibody, we compute the mean number of mutations over all its siblings; we do not aggregate over all children. Thus, for each method, pairwise diversity is represented by a distribution (each child generated by the method being a sample) rather than a single number.

For computing OASis humanness percentile scores, [Prihoda et al., 2022] we use the medium ($\geq$ 50% subjects) threshold. For each child antibody, we computed the germline frequency mutation delta as the sum over all mutated positions (relative to the seed) of the difference between the child residue's germline family frequency and the seed residue's germline family frequency at that position, using IMGT numbering-aware germline gene family frequency tables. For computing

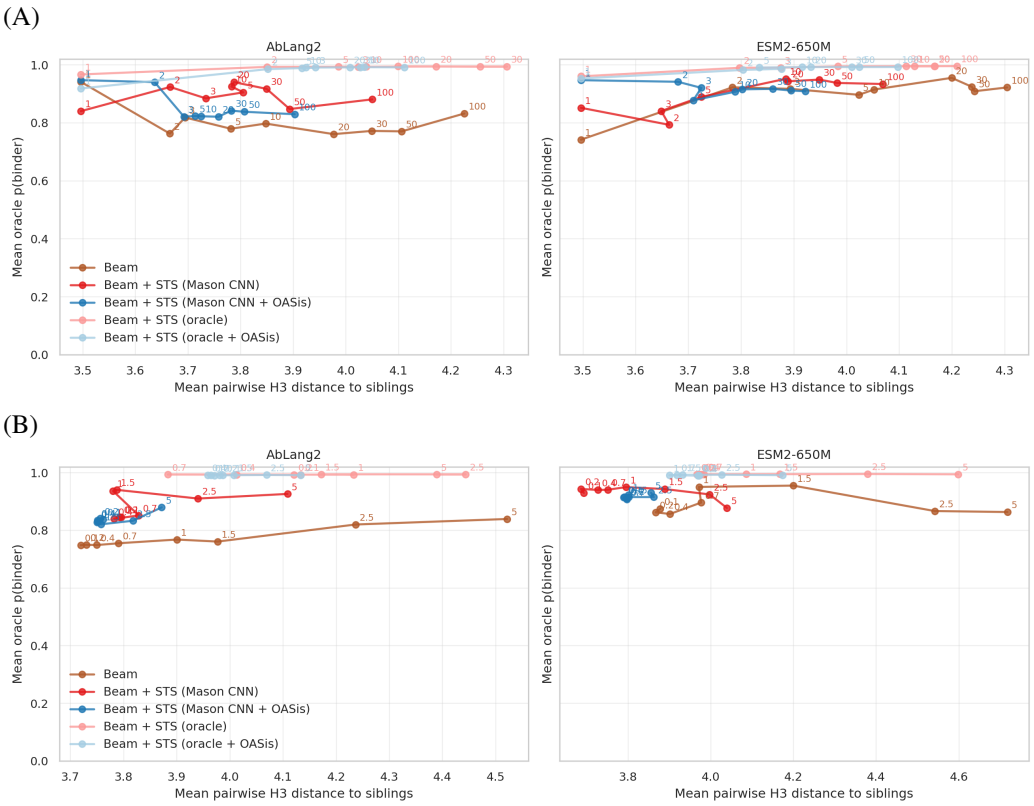


Figure 9: Results from sweeping beam size (A) and softmax temperature (B) on Trastuzumab experiment. Text annotations next to each marker denote the hyperparameter used for each experiment.

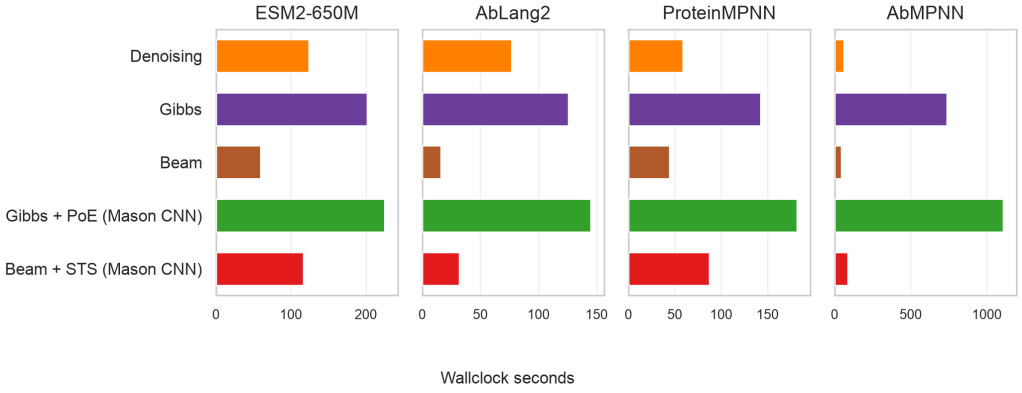


Figure 10: Wallclock time in seconds for Trastuzumab.

isoelectric point, we used Biopython v1.85 [Cock et al., 2009] with the concatenated VH and VL sequences. For the number of CDR mutations metric, for each child sequence we align [Dunbar and Deane, 2016] to the seed, and classify mutations using the Chothia [Chothia and Lesk, 1987] scheme.

Experimental results Results from an initial evaluation of sampling algorithms on AbLang2 are in Figure 11.

For the full in silico benchmarking results for humanness and germline bias are shown in Fig. 12A and Fig. 12B, respectively. The most noteworthy observations are that SAbDabMLM Gibbs pro-

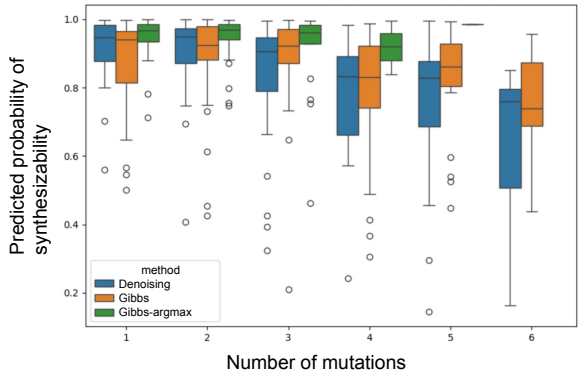


Figure 11: *In silico* predicted probability of synthesizability for AbLang2 generated sequences, as a function of number of edits.

duces sequences with low humanness. Meanwhile, CloneLM Beam, Sapiens Gibbs-argmax, and Amplify Beam have high germline bias. For isoelectric point in Fig. 12C, we observe that ESM2-650M Beam is the only method that simultaneously performed well in general, while also generating variants with low pI; thus it may be useful in scenarios where pI reduction is needed. For CDR mutation count in Fig. 12D, we see that SAbDabMLM Gibbs is similar to DiffAb+ in being biased to CDR-only mutations, despite not being intentionally designed to do so. This suggests that this behavior is due to biases in their shared training data. Conversely, we notice that pIgGen tends to produce mutations in framework regions.

A.9 Additional details on the in vitro experiment

A.9.1 Experimental setup

When generating variants with CDR-only mutations, we defined CDRs by the union of masks based on the Kabat [Kabat and Wu, 1971], Chothia [Chothia and Lesk, 1987], and IMGT [Lefranc et al., 2003] definitions.

Beam search was run with a beam size of 5, with the exception of the guidance methods which used a beam size of 20. All methods used a stochastic beam search temperature of 1.5. For unsupervised methods, the Gumbel-perturbed pseudo-log-likelihoods were used to rank sequences.

For supervised filtering, guidance, and ranking, we split the labeled 729 samples 60%-20%-20% into train-valid-test sets, and trained an ensemble of 5 models. We aligned sequences to a reference antibody sequence, and provided one-hot encodings of the aligned sequences as input features to the models. We trained CARP/Bytenet [Kalchbrenner et al., 2016, Yang et al., 2024] classification models to predict simultaneous synthesizability and binding success. The ensemble's models had test mean AUROC 0.9342 and standard deviation 0.0101. For filtering, we filtered generated sequences by ($p_{success} > 0.6$); for ranking, we sorted generated sequences by $p_{success}$. For NDS guidance and ranking, during beam search we combined AbLang2 pseudo-perplexity and supervised $p_{success}$, breaking NDS ties with the pseudo-perplexity; at ranking time, sequences were sorted by $p_{success}$. For STS guidance and ranking, during beam search we combined AbLang2 pseudo-perplexity (with $c_{\text{AbLang2}} = 1$) and supervised $p_{success}$ (with $c_{\text{supervised}} = 2$); at ranking time, sequences were sorted by $p_{success}$. Note that at ranking time, the ranker only had access to the top-1000 generated sequences, as sorted by the beam search criteria: for the unguided methods, these are the Gumbel-perturbed pseudo-log-likelihoods; for the guided methods, these are the aforementioned multiobjective scalarization criteria.

Before sending sequences to the wet-lab, designed sequences were filtered to have isoelectric point ≤ 9. Sequences were also filtered to exclude those introducing any of the following standard sequence liabilities relative to the seed: Asp-Pro cleavage sites, asparagine deamidation motifs (NG, NH, NN, NS, NT), aspartate isomerization motifs (DG, DD, DH, DS, DT), N-linked glycosylation

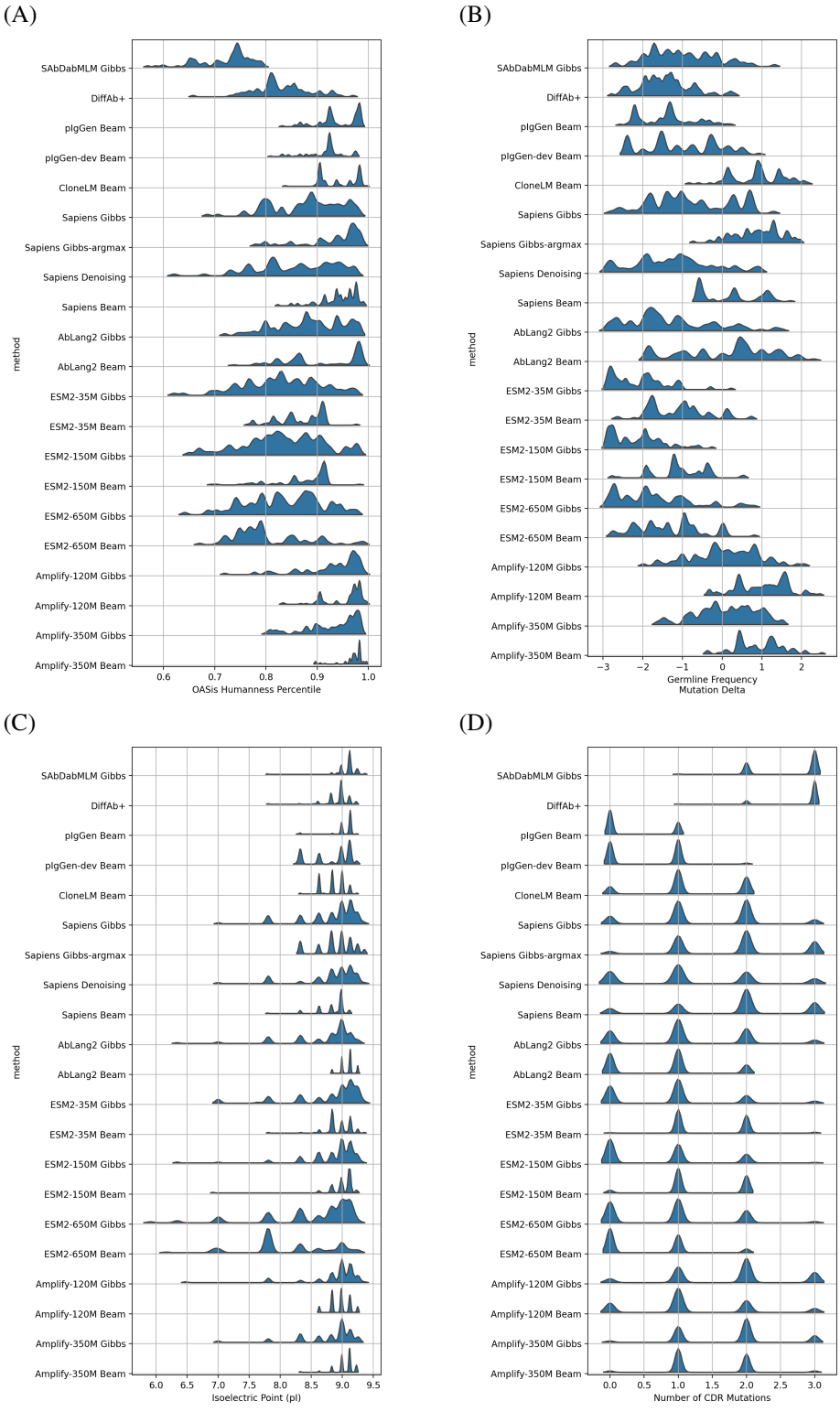


Figure 12: Additional *in silico* evaluation results for generated sequences: (A) OASis humanness percentile, (B) germline mutation bias, (C) isoelectric point, and (D) number of CDR mutations.

sequons (N-X-S/T, where X $\neq$ P), high-risk oxidation sites (exposed Met or Trp), and unpaired cysteine residues.

We used a seed with -log10 KD of 8.11 (i.e. 7.83 nM), yield of 42.8 ng per microliter, OASis percentile of 16.0% (14.5% VH, 17.4% VL), and overall pI of 6.14 (8.5 VH, 4.87 VL). For the OASis percentile, it is already OASis identity converted by BioPhi to a percentile against a reference set of therapeutic antibodies drawn from Thera-SAbDab (which includes includes murine and chimeric therapeutics). For pI, we downloaded Thera-SAbDab, then deduplicated and filtered to monospecific and VH:VL-paired, obtaining 914 clinical stage Fv sequences. This has min=4.60, p25=6.85, median=8.30, p75=8.73, and max=9.45.

A.9.2 In vitro binder classification

Binding kinetics were measured by bio-layer interferometry (BLI) and fit to a 1:1 binding model. Each sample was subjected to automated quality control to determine whether it produced a reliable binding measurement. Before kinetics-specific QC was applied, samples were checked for upstream protein production failures based on protein quantification metrics. A sample was failed if its protein concentration was below 50 ng/µL, or if its A260/A280 purity ratio fell outside the range of 0.5 to 0.85. Samples were classified as non-binders if the maximum binding response signal at the highest analyte concentration was less than 0.05 nm. We manually reviewed select BLI curves, verifying that these automated checks were indeed failing apparent non-binders, while passing apparent binders.

A.10 Limitations

The in vitro experimental results come from a single in vitro campaign with a single seed. These results may not generalize to other antibody campaigns, much less to non-antibody protein campaigns. Also, in the in vitro experiment, we only evaluated beam search with guidance, not Gibbs with guidance.

A.11 Computational resources used

Computations were performed either on a Nov 2023 MacBook Pro (Apple M3 Pro chip, 18GB memory), or on `g5.4xlarge` or `g5.12xlarge` AWS EC2 instances.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: See abstract and introduction.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: Limitations section is in Appendix.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [N/A]

Justification: No theory in paper.

Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: The proposed method and metrics are clearly described in the paper, that others may run the same analysis on their own protein optimization campaigns. We plan to release all code and models related to the Trastuzumab experiment upon acceptance.

   Guidelines:

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

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

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

Answer: [Yes]

Justification: We plan to release all code and models related to the Trastuzumab experiment upon acceptance.

Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: All details are in Appendix, to the best of our knowledge.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: Error bars were defined and used for the experimental results, where appropriate.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: "Computational resources used" section is in Appendix.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: No code violations, to the best of our knowledge.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [N/A]

    Justification: This work does not address societal impact because it is not necessary to do so for this paper.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [N/A]

    Justification: No data or models are being released.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: We cited the authors of public models that were used.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [Yes]

    Justification: We plan to release all code and models related to the Trastuzumab experiment upon acceptance.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: No Crowdsourcing and research with human subjects

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: No crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.

- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

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

    Answer: [N/A]

    Justification: LLMs were not involved in core method development.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.