PREFERENCE OPTIMIZATION OF PROTEIN LANGUAGE MODELS AS A MULTI-OBJECTIVE BINDER DESIGN PARADIGM

Pouria Mistani *
Aikium Inc.
Berkeley, CA 94704, USA
pouria@aikium.com

Venkatesh Mysore
Aikium Inc.
Berkeley, CA 94704, USA
venkatesh@aikium.com

ABSTRACT

We present a multi-objective binder design paradigm based on instruction fine-tuning and direct preference optimization (DPO) of autoregressive protein language models (pLMs). Multiple design objectives are encoded in the language model through direct optimization on expert curated preference sequence datasets comprising preferred and dispreferred distributions. We show the proposed alignment strategy enables ProtGPT2 to effectively design binders conditioned on specified receptors and a drug developability criterion. Generated binder samples demonstrate median isoelectric point (pI) improvements by $17\% - 60\%$.

1 INTRODUCTION

Peptides are an important class of biomolecules comprised of short strands of up to 50 amino acids. Designing peptide binders to specific protein targets with desirable therapeutic properties is a central problem in drug discovery (Fosgerau & Hoffmann, 2015). Beyond optimizing for binding affinity, peptide drug development processes require satisfying numerous other constraints imposed by physicochemical properties, their formulation characteristics, and pharmacodynamic influences on human subjects, among others. Recent progress in generative artificial intelligence has inspired novel strategies for designing protein binders, rooted in either structure-based or sequence-based protein representations. Filtering for designs that satisfy other objectives is performed *post facto*. A computational approach for generating peptides likely to satisfy multiple property constraints directly inferred from positive and negative examples is lacking; this work attempts to fill that lacuna.

Extending large language models (LLMs) for natural language processing (NLP) to biological sequences, protein language models (pLMs) are pre-trained on large scale evolutionary sequence data. Prominent foundation models include ESM2 (Lin et al., 2022) that is a BERT-style encoder transformer model, ProtGPT2 (Ferruz et al., 2022) and ProGen2 (Nijkamp et al., 2023) that are GPT-style decoder transformer models, and ProtT5 (Elnaggar et al., 2021) that is an encoder-decoder transformer model. These have been adapted for binder design models by fine-tuning these foundation models with different strategies. Examples of binder design models include PepMLM (Chen et al., 2023), DiffPALM (Lupo et al., 2023), IgLM (Shuai et al., 2023), and pAbT5 (Chu & Wei, 2023). These binder design models can be categorized by their formulation of the binder design problem as different NLP tasks. Three NLP tasks can be identified, (i) *text infilling* with masked language modeling such as PepMLM that is an encoder transformer based on fine-tuning ESM2 to reconstruct the fully masked binder region in a protein-peptide conjugated chain, (ii) *text generation* task with causal language models such as IgLM and ProtGPT2 that are decoder transformers based on GPT-2 (Radford et al., 2019), and (iii) *machine translation* task using sequence-to-sequence language modeling such as pAbT5 that generates the heavy/light chain given their chain pairing partner in antibody sequences by fine-tuning the decoder stage in ProtT5.

On the other hand, structure-based models attempt to reason about binding and other properties by carefully constructing in three dimensions bound poses of the protein-peptide complex (Chang et al., 2024). Despite being hindered by limited availability of solved protein-peptide complex structures

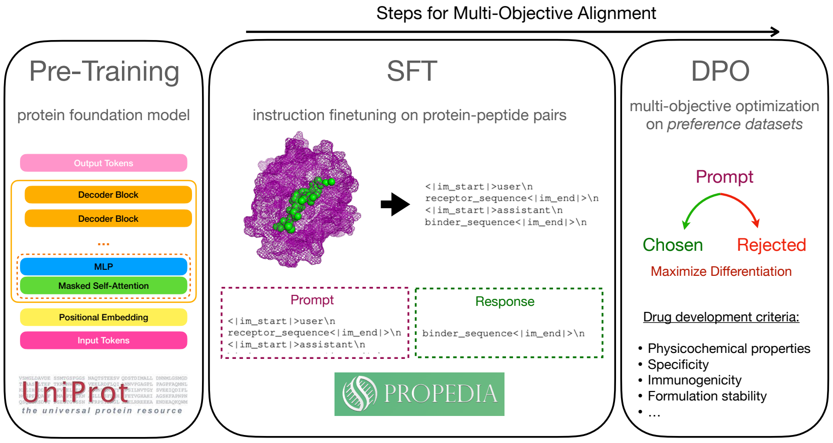
*Corresponding author.

and the computational cost of molecular dynamics simulations, good progress has been made in structure-informed peptide design(Kosugi & Ohue, 2022; gou, 2023; Bryant & Elofsson, 2023). More recently, diffusion models in the structure space (Anand & Achim, 2022; Watson et al., 2023) as well as the sequence space (Alamdari et al., 2023; Gruver et al., 2023) are being adapted for peptide design (Xie et al., 2023; Wang et al., 2024).

Drug discovery and development is a multi-objective optimization process. Beyond binding affinity, numerous other factors need to be considered for therapeutic development such as expressibility, synthesizability, stability, immunogenicity, solubility and bioavailability. Our goal is to develop a framework for binder design beyond binding affinity, where downstream properties and experimental heuristics from human experts can be readily incorporated in a multi-objective optimization framework. Interestingly, techniques such as Reinforcement Learning from Human Feedback (RLHF) (Christiano et al., 2017; Bai et al., 2022) and Direct Preference Optimization (DPO) (Rafailov et al., 2023) can instill desired behaviors in the responses generated by large language models; *e.g.*, see (Park et al., 2023) for unconditional small molecule generation. In this study, we optimize autoregressive pLMs to capture diverse preferred and undesired protein sequence distributions, while conditioned on target receptor sequences. This approach enables development of computational frameworks for multi-objective drug design. We use positive and negative data distributions for protein-peptide binding affinity as well as peptide isoelectric points (pI) to show that the model can learn to generate novel sequences that simultaneously respect these different objectives.

2 Methods

We propose an alignment method to transform pre-trained unconditional protein sequence models ($p(s)$), that autoregressively sample sequences (s) from underlying data distribution ($\mathcal{D}$), to conditional probability models ($p(s|r;c)$) that given a target receptor (r) sample binders that satisfy constraints (c) encoded by preference datasets compiled from experiments and domain experts.



```
<|im_start|>user\n
receptor_sequence<|im_end|>\n
<|im_start|>assistant\n
binder_sequence<|im_end|>\n
```

```
<|im_start|>user\n
receptor_sequence<|im_end|>\n
<|im_start|>assistant\n
```

```
binder_sequence<|im_end|>\n
```

Figure 1: Alignment method for multi-objective optimization of favorable binders

We perform a two step instruction fine-tuning, as in (Ouyang et al., 2022), see figure 1: (i) we instill receptor-binder *chat-templates* given in table 1 through supervised fine-tuning (SFT), and (ii) we optimize the fine-tuned model to promote preferred binders over dispreferred ones. In this work, we curate preference datasets to induce high specificity binders with favorable isoelectric point values (*i.e.*, pH at which the peptide is electrically neutral). Specifically, for ease of illustration, we demonstrate how to nudge the model to generate peptides with a higher pI.

2.1 Supervised fine-tuning (SFT) for protein-peptide binders

We fine-tuned ProtGPT2 for instruction tasks following the OpenAI chatML template (OpenAI, 2023), see table 1. We fine-tuned all linear layers (including embedding layers, attention blocks and MLPs) with QLoRA optimization (Dettmers et al., 2023) for 24 epochs; see appendix A for more

Table 1: OpenAI ChatML template for receptor-binder instruction fine-tuning

Message template **Message segments**

```
<|im_start|>user\n                          |                         |
receptor_sequence<|im_end|>\n               | -> (generation prompt)  |
<|im_start|>assistant\n                     |                         | -> (SFT message)
binder_sequence<|im_end|>\n                   -> (completion)         |
```

details. The model learns to generate binders when prompted by the generation template for a given receptor. The sequences follow FASTA convention of inserting next-line special character '\n' after every 60 residues. Note that ProtGPT2 uses the Byte Pair Encoding (BPE) tokenizer (Sennrich et al., 2015) with $50,257$ vocabulary size (recall we added two extra tokens) that contains higher order oligomers up to 9 residues long. This is in contrast to common pLMs such as ESM2 with only 33 vocabulary size. Supporting more tokens primes the foundation model for more complex design tasks such as AI design agents from textual descriptions.

Fine-tuning is performed using a causal language modeling objective. Each receptor-binder sequence pair is represented in the format of table 1, then tokenized into a set of symbols $s = (t_1, \cdots, t_n)$. Assuming probability of next token depends on preceding tokens, the total probability of a sequence pair, $s^{(k)}$, is $p(s^{(k)}) = \Pi_{i=1}^{n} p(t_i|t_1, \cdots, t_{i-1})$. Therefore, we estimate a pLM to predict conditional probabilities, $p \sim \pi_\theta(t_i|t_{<i})$, by minimizing the model negative log-likelihoods:

$$\mathcal{L}_{SFT}(\pi_\theta) = \mathbb{E}_{(t_i \in s; s \sim \mathcal{D})}\big[-\log \pi_\theta(t_i|t_{<i})\big]$$

2.2 Direct preference optimization (DPO) for multi-objective design

Let π_θ be the pre-trained protein language model. In the first phase, the model is subjected to supervised fine-tuning (SFT) for the instruction task using a high quality receptor-binder dataset (x, y), to obtain a reference model $\pi_{\text{ref}} = \pi^{\text{SFT}}$. In the second phase, the reference model is prompted with target proteins, x, to sample binder pairs, $y_w, y_l \sim \pi_{ref}(\cdot|x)$, where y_w is preferred and y_l is dispreferred by some criteria. In practice, we use carefully curated offline preference datasets $\mathcal{D} = \{x^{(i)}, y_w^{(i)}, y_l^{(i)}\}_{i=1}^{N}$. The preferences are assumed to be sampled from the Bradley-Terry (BT) (Bradley & Terry, 1952) reward model $r^*(y, x)$:

$$p^*(y_1 \succ y_2|x) = \sigma\big(r^*(x, y_1) - r^*(x, y_2)\big) \tag{1}$$

In RLHF, a parameterized reward model $r_\phi(y, x)$ is used to provide feedback for optimizing the language model. The reward model is explicitly estimated using the negative log-likelihood loss $\mathbb{E}_{(\text{x}, \text{y}_w, \text{y}_l) \sim \mathcal{D}}[-\log \sigma(r_\phi(x, y_w) - r_\phi(x, y_l))]$ on the preference data, which is then used to optimize the optimal policy $\pi^* = \arg\max_{\pi_\theta} \mathbb{E}_{x \in \mathcal{D}, y \in \pi_\theta}[r_\phi(x, y)] - \beta D_{KL}\big(\pi_\theta(y|x)||\pi_{ref}(y|x)\big)$. This is commonly achieved using the PPO algorithm (Schulman et al., 2017). In DPO, the reward function is explicitly expressed in terms of its optimal policy $r^*(x, y) = \beta \log \dfrac{\pi^*(y|x)}{\pi_{ref}(y|x)} + \beta \log Z(x)$. Thus, the optimal model can be expressed only in terms of the optimal policy and the reference policy, bypassing the need for estimating the reward model. Therefore, the DPO loss is directly minimizing the negative log-likelihood of the preference model in equation 1:

$$\mathcal{L}_{DPO}(\pi_\theta; \pi_{ref}) = \mathbb{E}_{(\text{x}, \text{y}_w, \text{y}_l) \sim \mathcal{D}}\left[-\log \sigma\left(\beta \log \frac{\pi_\theta(y_w|x)}{\pi_{ref}(y_w|x)} - \beta \log \frac{\pi_\theta(y_l|x)}{\pi_{ref}(y_l|x)}\right)\right] \tag{2}$$

where β determines information retention from the reference model.

2.3 Preference datasets: specificity & isoelectric points

For each target protein we can provide several preference datasets. The preference datasets follow the format given in table 2. Rejection criteria model unfavorable properties (*e.g.*, unfavorable physicochemical properties). Rejected samples are a strong lever for instilling expert heuristics or information about downstream properties in drug development into the protein language model. In this work, we define two types of dispreferred sequences (1) to enhance target specificity (*i.e.*, binders are specific to their cognate receptors and not to other target receptors from the sample), and (2) to avoid undesirably low pI, characterized by an excessive number of negatively charged residues.

Table 2: Format of preference data for DPO

<table>
<tr><th>Prompt</th><th>Chosen</th><th>Rejected</th><th>Criteria</th></tr>
<tr><td><|im_start|>user\n
LRGLSEDTLEQLYALGFNQ...<|im_end|>\n
<im_start>assistant\n</td><td>TGVALTPPS<|im_end|>\n</td><td>CRGCX<|im_end|>\n</td><td>affinity/specificity</td></tr>
<tr><td><|im_start|>user\n
LRGLSEDTLEQLYALGFNQ...<|im_end|>\n
<im_start>assistant\n</td><td>TGVALTPPS<|im_end|>\n</td><td>TGVDLTEPS<|im_end|>\n</td><td>isoelectric point</td></tr>
</table>

Receptor-binder dataset We followed (Chen et al., 2023) to compile protein-peptide pairs from PepNN (Abdin et al., 2022) and Propedia (Martins et al., 2023) datasets. We filtered protein-peptide pairs with cutoff lengths of 500 and 50, respectively. We applied a homology filter with 80% threshold to remove redundancies. This process led to $9,439$ pairs, involving 6,570 unique proteins and 7,557 unique peptides.

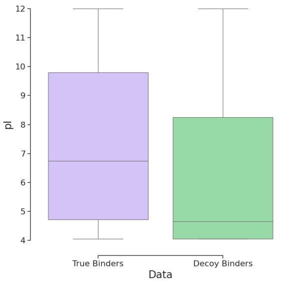


Figure 2: Statistics of isoelectric points in validation data

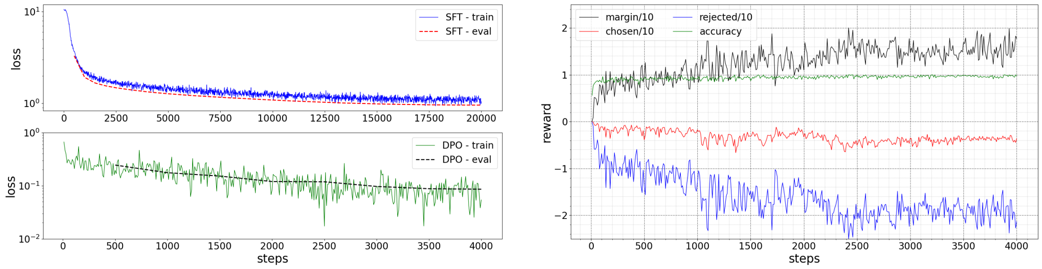
Specificity preference dataset We clustered the proteins and peptides separately using Mmseqs (Steinegger & Söding, 2017), using a minimum sequence identity of 0.8 and artificially constructed dispreferred protein-peptide pairs by matching ones from distant groups. A true binder peptide was assigned as a decoy dispreferred peptide to a new protein that was in a different protein-cluster. Further, to the extent possible, it was ensured that this reassigned peptide was not in the same peptide-cluster as the true binder peptides of the protein it was getting paired with.

Isoelectric point preference dataset Charges on a peptide contribute to its pI (more negative charges lower its pI). We construct a dispreferred peptide from a true binding peptide by mutating 20% of the residue positions to a negatively charged amino acid; the positions and the specific acid (glutamic or aspartic) are chosen randomly. Thus, for each *true binder* we added a *decoy binder* with lower pI, see figure 2.

For each protein, each true binder peptide was paired with every dispreferred decoy peptide (be it a true binder of a distant protein or a mutated form of another true binder of the same protein) to construct one training example for the DPO phase of training. This enrichment with preference datasets led to $66,898$ triplets of proteins, binders, and decoys. We held out $3,345$ triplets for testing.

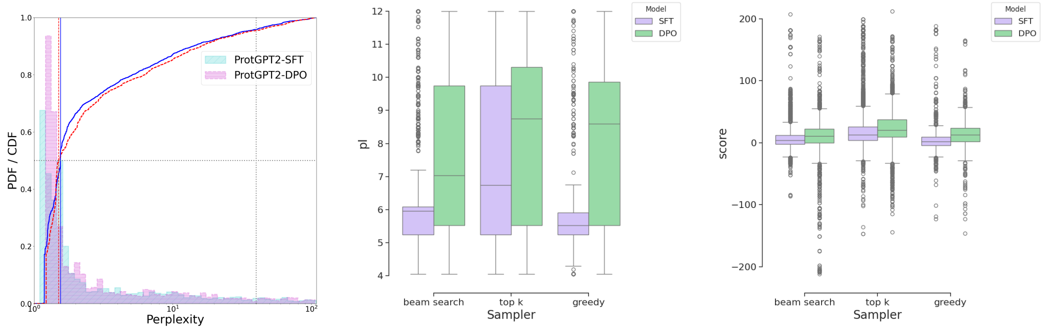
3 Results

Training metrics The SFT stage instills receptor-binder completions into ProtGPT2 by training on the preferred binders, reaching a local minimum. At this state the model has maximal log probabilities for the *chosen* samples, and any further optimization during DPO for discriminating the *rejected* samples will inevitably degrade the log probability of the chosen data, note this is an expected outcome for any multi-objective optimization. The DPO optimization objective is maximizing the *preference reward* (subject to KL divergence from the fine-tuned model), but the SFT objective is to maximize the chosen log probabilities. Figure 3(a) illustrates that both losses decrease to the local minimum, and figure 3(b) shows reward metrics improve during DPO where both chosen and rejected reward log-probabilities decrease while the rejected reward decreases much faster, leading to improving margin and accuracy. Detailed definition of these metrics is given in appendix B. Overall, after 24 epochs of SFT and, subsequently, 1 epoch of DPO we achieved an accuracy of 97.5% and a reward margin of 17.7. On a held-out test data accuracy and margin are 97% and 15.3, respectively.
Binder perplexity Perplexity (Jelinek et al., 1977) is commonly used for evaluating autoregressive models. Sequence perplexity (PPL) is the exponentiated average of model negative log-likelihoods for tokens in the sequence conditioned on their previous tokens. We used beam search, top-k, and greedy sampling to generate up to 3 binders for each receptor in our test data, see appendix C. Figure 4(a) compares PPLs of both SFT and DPO models and shows 50% of binders exhibit a PPL less than ~ 1.5 and 90% are below 40. Moreover, we observe DPO does not significantly deteriorate PPLs.



(a) Losses during training (b) DPO rewards (chosen, rejected, margin, accuracy)

Figure 3: Training metrics for SFT and DPO. See appendix B for definition of these metrics.



(a) Perplexities with beam-search (b) Isoelectric points, pI (c) Alignment scores

Figure 4: Generated binders by both SFT and DPO have low perplexities (left). DPO significantly improves pI (middle) and alignment scores (right)

Binder pI Figure 4(b) shows DPO enhances the median pI values by a factor of 1.2, 1.3, or 1.5 over SFT generations using beam-search, top-K, or greedy samplers, respectively. Interestingly, we observe after DPO 50% of binders undergo pI improvements by a factor of 2.

Binder alignment score To examine similarity of generated binders with distributions of ground truth binders in the validation data, for each generated binder we computed the best alignment score to all positive true binders of receptors in the same cluster as its cognate protein. Figure 4(c) illustrates alignment score enhancements by DPO over the baseline SFT designs. Overall, all pI and score improvements are strong and in the expected direction in all three sampling strategies.

4 Conclusions

To the best of our knowledge, this is the first incorporation of DPO in protein language models for generating peptides with desirable physicochemical properties that bind to a given target protein. While the work describes adapting DPO into pLMs for peptide design, the approach is equally applicable to protein or small molecule design. Rather than having a down-stream classifier or regressor for property prediction and acceptance / rejection of a designed molecule, the framework proposed in this work affords a streamlined way to incorporate preferences ahead of time. Importantly, it opens the way to utilizing the vast amount of negative data that would have previously been deemed irrelevant during pre-training or fine-tuning. Additionally, the model can be shown dispreferred examples labeled unacceptable for reasons that may not be quantifiable by a numerical threshold on a single property (*e.g.*, expert opinion, expression failure, synthetic feasibility), expanding the breadth of considerations. We anticipate that this approach will be integrated into peptide, protein and small molecule design projects in different settings, leading to an increase in efficiency of the drug discovery process by increasing the likelihood of hit molecules being translated into therapeutically viable lead molecules.

References

In silico evolution of protein binders with deep learning models for structure prediction and sequence design. *bioRxiv [Preprint]*, 2023.

Osama Abdin, Satra Nim, Han Wen, and Philip M Kim. Pepnn: a deep attention model for the identification of peptide binding sites. *Communications biology*, 5(1):503, 2022.

Sarah Alamdari, Nitya Thakkar, Rianne van den Berg, Alex Xijie Lu, Nicolo Fusi, Ava Pardis Amini, and Kevin K Yang. Protein generation with evolutionary diffusion: sequence is all you need. *bioRxiv*, pp. 2023–09, 2023.

Namrata Anand and Tudor Achim. Protein structure and sequence generation with equivariant denoising diffusion probabilistic models. *arXiv preprint arXiv:2205.15019*, 2022.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*, 2022.

Ralph Allan Bradley and Milton E Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.

Patrick Bryant and Arne Elofsson. Peptide binder design with inverse folding and protein structure prediction. *Communications Chemistry*, 6(229), 2023.

Liwei Chang, Arup Mondal, Bhumika Singh, Yisel Martínez-Noa, and Alberto Perez. Revolutionizing peptide-based drug discovery: Advances in the post-alphafold era. *WIREs Computational Molecular Science*, 14(1):e1693, 2024. doi: https://doi.org/10.1002/wcms.1693. URL https://wires.onlinelibrary.wiley.com/doi/abs/10.1002/wcms.1693.

Tianlai Chen, Sarah Pertsemlidis, Venkata Srikar Kavirayuni, Pranay Vure, Rishab Pulugurta, Ashley Hsu, Sophia Vincoff, Vivian Yudistyra, Lauren Hong, Tian Wang, et al. Pepmlm: Target sequence-conditioned generation of peptide binders via masked language modeling. *ArXiv*, 2023.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017.

Simon Chu and Kathy Wei. Generative antibody design for complementary chain pairing sequences through encoder-decoder language model. In *NeurIPS 2023 Generative AI and Biology (GenBio) Workshop*, 2023.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms. *arXiv preprint arXiv:2305.14314*, 2023.

Ahmed Elnaggar, Michael Heinzinger, Christian Dallago, Ghalia Rehawi, Yu Wang, Llion Jones, Tom Gibbs, Tamas Feher, Christoph Angerer, Martin Steinegger, et al. Prottrans: Toward understanding the language of life through self-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 44(10):7112–7127, 2021.

Angela Fan, Mike Lewis, and Yann Dauphin. Hierarchical neural story generation. *arXiv preprint arXiv:1805.04833*, 2018.

Noelia Ferruz, Steffen Schmidt, and Birte Höcker. Protgpt2 is a deep unsupervised language model for protein design. *Nature communications*, 13(1):4348, 2022.

Keld Fosgerau and Torsten Hoffmann. Peptide therapeutics: current status and future directions. *Drug Discovery Today*, 20(1):122–128, 2015. ISSN 1359-6446. doi: https://doi.org/10.1016/j.drudis.2014.10.003. URL https://www.sciencedirect.com/science/article/pii/S1359644614003997.

Nate Gruver, Samuel Stanton, Nathan C Frey, Tim GJ Rudner, Isidro Hotzel, Julien Lafrance-Vanasse, Arvind Rajpal, Kyunghyun Cho, and Andrew Gordon Wilson. Protein design with guided discrete diffusion. *arXiv preprint arXiv:2305.20009*, 2023.

Fred Jelinek, Robert L Mercer, Lalit R Bahl, and James K Baker. Perplexity—a measure of the difficulty of speech recognition tasks. *The Journal of the Acoustical Society of America*, 62(S1): S63–S63, 1977.

Takatsugu Kosugi and Masahito Ohue. Solubility-aware protein binding peptide design using alphafold. *Biomedicines*, 10(7), 2022. ISSN 2227-9059. doi: 10.3390/biomedicines10071626. URL `https://www.mdpi.com/2227-9059/10/7/1626`.

Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, 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*, 2022:500902, 2022.

Umberto Lupo, Damiano Sgarbossa, and Anne-Florence Bitbol. Pairing interacting protein sequences using masked language modeling. *bioRxiv*, pp. 2023–08, 2023.

Pedro Martins, Diego Mariano, Frederico Chaves Carvalho, Luana Luiza Bastos, Lucas Moraes, Vivian Paixão, and Raquel Cardoso de Melo-Minardi. Propedia v2. 3: A novel representation approach for the peptide-protein interaction database using graph-based structural signatures. *Frontiers in Bioinformatics*, 3:1103103, 2023.

Erik Nijkamp, Jeffrey A Ruffolo, Eli N Weinstein, Nikhil Naik, and Ali Madani. Progen2: exploring the boundaries of protein language models. *Cell Systems*, 14(11):968–978, 2023.

OpenAI. Chatml documentation. `https://github.com/openai/openai-python/blob/release-v0.28.0/chatml.md`, 2023. Accessed: February 5, 2024.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35: 27730–27744, 2022.

Ryan Park, Ryan Theisen, Navriti Sahni, Marcel Patek, Anna Cichońska, and Rayees Rahman. Preference optimization for molecular language models. *arXiv preprint arXiv:2310.12304*, 2023.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *arXiv preprint arXiv:2305.18290*, 2023.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*, 2015.

Richard W Shuai, Jeffrey A Ruffolo, and Jeffrey J Gray. Iglm: Infilling language modeling for antibody sequence design. *Cell Systems*, 14(11):979–989, 2023.

Martin Steinegger and Johannes Söding. Mmseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nature biotechnology*, 35(11):1026–1028, 2017.

Yongkang Wang, Xuan Liu, Feng Huang, Zhankun Xiong, and Wen Zhang. A multi-modal contrastive diffusion model for therapeutic peptide generation, 2024.

Joseph L Watson, David Juergens, Nathaniel R Bennett, Brian L Trippe, Jason Yim, Helen E Eisenach, Woody Ahern, Andrew J Borst, Robert J Ragotte, Lukas F Milles, et al. De novo design of protein structure and function with rfdiffusion. *Nature*, 620(7976):1089–1100, 2023.

Xuezhi Xie, Pedro A Valiente, Jisun Kim, and Philip M Kim. Helixdiff: Hotspot-specific full-atom design of peptides using diffusion models. In *Machine Learning for Structural Biology Workshop, NeurIPS*, 2023.

A Training parameters

ProtGPT2 is trained with one special token `<|endoftext|>`, however to accommodate for the instruction tuning task we introduced two extra special tokens `<|im_start|>` and `<|im_end|>` to signify begin and end of prompts and responses. We repurposed `<|endoftext|>` as the padding token. These changes require retraining the embedding layers at the input of the network and at the model head due to enlarged embedding dimensions. Our model has 774 million parameters.

We list the hyper-parameters used for training and model architecture in table 3.

Table 3: Hyper-parameters for training and model architecture.

Model Architecture	Decoder transformer with 36 attention blocks and 774 million parameters. (ProtGPT2 model has 738 million parameters, our model is larger due to added tokens). Embedding dimensionality is $1,280$; input/output space is $57,259$ tokens.
Dataset	Synthesized from $9,439$ unique protein-peptide pairs from the Propedia and PepNN databases. Compiled $66,898$ sequence triplets of "protein-peptide-decoy" for DPO.
Hyperparameters	Train/eval batch size per device 2/8. 501 warm-up steps; linear scheduler with learning rate 5×10^{-4}; AdamW optimizer. QLoRA with $\alpha = 16$, rank $r = 16$, dropout is 0.05.
Training Duration **Hardware** **Training Time**	Trained for $20,000$ SFT steps, $4,000$ DPO steps. 2 NVIDIA RTX A6000 GPUs. ~ 7.5 hours (24.36 epochs) for instruction fine-tuning, and ~ 1.3 hour for DPO (1 epoch).
Initialization	Initialized with pre-trained weights from ProtGPT2 trained on 50 million sequences from UniProt database.

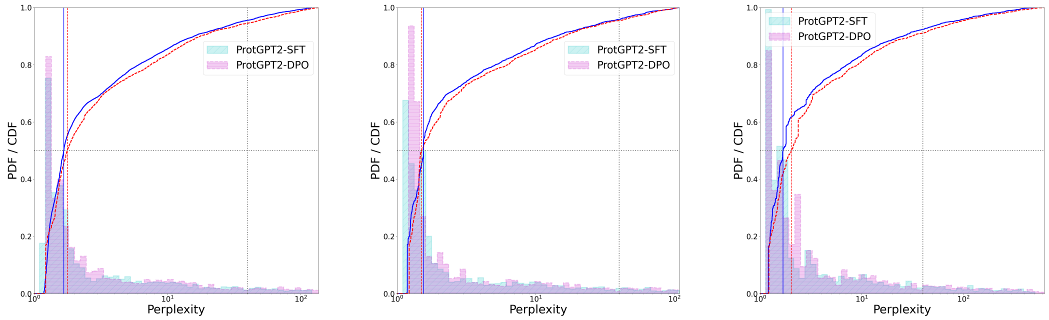
B DPO metrics

Below are definitions for the reported reward metrics for DPO:

- $chosen\ reward = \mathbb{E}_{(\mathrm{x},\mathrm{y}_w,\mathrm{y}_l)\sim\mathcal{D}}\big[\beta \log \frac{\pi_\theta(y_w|x)}{\pi_{ref}(y_w|x)}\big]$, quantifying the mean difference between the log probabilities of the policy model and the reference model on the chosen responses.
- $rejected\ reward = \mathbb{E}_{(\mathrm{x},\mathrm{y}_w,\mathrm{y}_l)\sim\mathcal{D}}\big[\beta \log \frac{\pi_\theta(y_l|x)}{\pi_{ref}(y_l|x)}\big]$: quantifying the mean difference between the log probabilities of the policy model and the reference model on the rejected responses.
- $accuracy$: mean frequency of chosen rewards being greater than the rejected rewards.
- $margin$: the mean difference between the chosen and rejected rewards.

C Sampling strategies

Different sampling strategies generate sequences with varying levels of model uncertainty. Here we experimented influences of different sampling strategies on induced sequence perplexities. A higher perplexity implies higher model uncertainty. We experimented with different sampling strategies including top-k (Fan et al., 2018) (with top-p), greedy, and beam search. Beam-search generated lowest binder perplexities when conditioned on receptor sequences; see figures 5.



(a) Top-K sampling with top 3 sequences per each prompt, with $\mathrm{top\_k} = 950$ and $\mathrm{top\_p} = 0.85$ and $\mathrm{temperature} = 0.7$. Median perplexities are $\mathrm{SFT} = 1.67$ and $\mathrm{DPO} = 1.77$.

(b) Beam-search sampling with top 3 sequences out of 20 beams per each prompt. Median perplexities are $\mathrm{SFT} = 1.67$ and $\mathrm{DPO} = 1.50$.

(c) Greedy sampling. Median perplexities are $\mathrm{SFT} = 1.67$ and $\mathrm{DPO} = 1.90$.

Figure 5: Probability (and cumulative) distribution functions for perplexities computed with different sampling strategies. Receptors from a held-out validation set were used to prompt the models for binder designs.