Extrapolative Protein Design through Triplet-based Preference Learning

Mostafa Karimi [1] **Sharmi Banerjee** [1] **Tommi Jaakkola** [2] **Bella Dubrov** [1] **Shang Shang** [1] **Ron Benson** [1]

Abstract

Extrapolative protein design is a crucial task for automated drug discovery to design proteins with higher fitness than what has been seen in training (eg. higher stability, tighter binding affinity, etc.). The current state-of-the-art methods assume that one can safely steer protein design in the extrapolation region by learning from pairs alone. We hypothesize that (1) noisy pairs do not accurately approximate gradient to improve fitness (2) it is challenging for the models to learn higher order relationships among designs (triplets, etc) from noisy pairs alone. Motivated by the success of alignment in large language models, we have developed an extrapolative protein design via triplet-based preference learning for both better approximation of gradient and directly modeling ranks of triplets fitness. We evaluated our model's performance in designing AAV and GFP proteins and demonstrated that the proposed framework significantly improves the generative models' effectiveness in extrapolation tasks.

1. Introduction

We focus on the challenging but crucial task of extrapolative protein design (Chan et al., 2021; Padmakumar et al., 2023; Lee et al., 2023) involving creation of novel sequences with enhanced fitness that surpasses the training distribution for e.g. designing antibodies with greater stability or stronger binding affinity. Extrapolation is challenging for deep neural networks as they are primarily trained to recognize patterns within the range of the training data (Xu et al., 2020). Existing extrapolative protein design models learn from the fitness ranking of protein pairs to extrapolate to higher fitness beyond training data. The gradient direction is approximated through differences in fitness between protein pairs. Current state-of-the-art models learn the ranking through contrastive discriminatory objective (Chan et al., 2021), token-level machine translation (Padmakumar et al., 2023) and Bradley-Terry (BT) model (Bradley & Terry, 1952) with maximum likelihood objective (Lee et al., 2023). The primary limitations are that the noisy approximation of gradient direction from pairs is not enough to steer protein generation into extrapolation region and that higher order ranking among protein sequences (triplets etc.) cannot be easily learned from the noisy pairs.

To address these limitations, and drawing inspiration from the recent success of human preference learning (Christiano et al., 2017; Rafailov et al., 2023) to guard LMs against harmful and undesired text generation, we propose a novel triplet-based preference learning method for extrapolative protein design, aiming to better approximate the gradient direction through triplewise ordering. We have approximated the triplewise relationship $f(x_1, x_2, x_3) \approx g(x_1, x_2) - g(x_3, x_2)$ through Bradley-Terry (BT) similarly as in direct preference optimization (DPO) (Rafailov et al., 2023). Our main objectives are to improve the approximation of gradient direction using triplets and guide language models towards higher fitness in the extrapolation region while preventing the generation of lower fitness sequences through preference learning. The proposed model follows the "do no harm" principle during extrapolation and provides a simple yet effective method for modeling triplewise relationships that can be readily applied to any extrapolative biological design problem.

2. Related works

All existing extrapolative protein design models have an inherent underlying assumption that extrapolation can be sufficiently learned through pairwise ranking of protein fitness. (Chan et al., 2021) developed a contrastive approach of ranking pairs using a discriminator of the latent space and extrapolating proteins by traversing through it. Padmakumar et al. (2023) proposed a local editor for translating sequences with low fitness to sequences with slightly higher fitness through machine translation. Recently, (Lee et al., 2023) modeled the ranked pairs through Bradley-Terry (BT) model via maximum likelihood objective. Aligning language model's output with human feedback has improved their abilities in following instructions (Ouyang et al., 2022) and trans-

[1]Amazon, Seattle, WA, USA [2]Massachusetts Institute of Technology, MIT, Cambridge, MA, USA. Correspondence to: Mostafa Karimi <mkarimii@amazon.com>.

Published at ICML 2024 Workshop on Foundation Models in the Wild.

lation (Kreutzer et al., 2018). LLM alignment originated from the seminal work (Christiano et al., 2017) through reinforcement learning with human feedback (RLHF). Training RLHFs is challenging due to the training instabilities, reward hacking and catastrophic forgetting (Peng et al., 2023). Recently, there has been a momentum towards closed-form and direct optimization of offline preferences such as direct preference optimization (DPO) (Rafailov et al., 2023). Direct preference models not only perform at par with RLHFs but also are simpler to implement and computationally efficient given their single-stage training strategy.

3. Methods

3.1. Problem Definition

Let's assume there is a supervised dataset $\mathrm{D} = \{(\mathbf{x}^n, y^n)\}_{n=1}^N$ with N samples where $\mathbf{x}^n = (x_1^n, \cdots, x_L^n)$ is nth protein sequence with length L and y^n is its corresponding fitness value (i.e. stability, binding affinity, etc). Let's assume the fitness value y in dataset D is bounded $y \in [y_{\min}, y_{\max}]$. We define this region as *training region* and try to generate sequences with fitness value $y_{\text{gen}} > y_{\max}$ or $y_{\text{gen}} < y_{\min}$ which is defined as *extrapolation region*.

3.2. Overview

The core concept behind the proposed method is to gradually learn the higher order relationships among ranked proteins. Starting with an auto-regressive unconditional pLM such as Prot-T5-XL (Elnaggar et al., 2021) that is trained on unsupervised data to model $\mathbf{x} \sim P_\theta(.)$ where $\mathbf{x}$ is generated protein sequence. Inspired by ICE model (Padmakumar et al., 2023), we trained a local editor with the desired direction (e.g. increasing the binding affinity) to learn the first order relationship among ranked proteins (approximating desired gradient direction through pairs). The model learns to generate $\mathbf{x}_2 \sim P_\theta(.|\mathbf{x}_1)$ where the fitness of $\mathbf{x}_2$ (designed sequence) is expected to be better than $\mathbf{x}_1$ (starting sequence). Inspired by direct preference optimization (DPO) (Rafailov et al., 2023) we aligned the models based on triplets by directly optimizing on newly created preferences. With this alignment, the model updates its belief of gradient direction from triplewise relationships. The overall schematic of the proposed method is illustrated in Figure 1.

3.3. Local editing through pairs

Given a supervised dataset D, we trained an scorer function f_s to predict the fitness of a query sequence. We expect f_s to perform well on training region and perform poorly on the extrapolation region since it has not seen these fitness during its training. Then, following (Padmakumar et al., 2023) we generated perturbed sequences by masking-infilling starting from the training sequences (seeds). Scorer function f_s is utilized to assess whether the newly generated pair (seed, sequence) has small but meaningful improvement toward desired direction. Dataset $\mathrm{D}_{\text{pair}} = \{(\mathbf{x}^m, \mathbf{z}^m)\}_{m=1}^M$ with M samples where $f_s(\mathbf{x}^m) < f_s(\mathbf{z}^m)$ if increasing fitness is desired and vice versa. Finally, we fine-tuned Prot-T5-XL model (Elnaggar et al., 2021) through MLE in an auto-regressive manner to predict the next amino acid: $\mathrm{P}_{\text{pair}}(\mathbf{z}|\mathbf{x}) = \prod_{i=1}^L \mathrm{P}(z_i|\mathbf{z}_{<i}, \mathbf{x})$.

3.4. Preference learning through triplets

To better approximate the gradient direction toward improved fitness in the extrapolation region and directly model higher order relationship among proteins, we created a preference dataset of size K based on triplets $\mathrm{D}_{\text{triplet}} = \{(\mathbf{x}_{\text{prompt}}^k, \mathbf{x}_w^k, \mathbf{x}_l^k)\}_{k=1}^K$ where $\mathbf{x}_{\text{prompt}}$ is seed sequence, $\mathbf{x}_w$ is the *desired* response and $\mathbf{x}_l$ is the *undesired* response. We are interested in increasing fitness by moving from $\mathbf{x}_{\text{prompt}}$ toward $\mathbf{x}_w$ where $f_s(\mathbf{x}_{\text{prompt}}) < f_s(\mathbf{x}_w)$ while guarding it against sequences with same or worse finesses (undesired ones). Therefore, we would create the following preference datasets (i) *Don't go backward*: triplets should satisfy the following order $f_s(\mathbf{x}_l) < f_s(\mathbf{x}_{\text{prompt}}) < f_s(\mathbf{x}_w)$ (ii) *Don't get stuck at the same fitness*: triplets should satisfy the following order $f_s(\mathbf{x}_l) \approx f_s(\mathbf{x}_{\text{prompt}}) < f_s(\mathbf{x}_w)$.

Inspired by DPO (Rafailov et al., 2023), we model triplewise relationship among proteins via Bradley-Terry (BT) model (Bradley & Terry, 1952). Formally, for a given triplet of $(\mathbf{x}_{\text{prompt}}, \mathbf{x}_w, \mathbf{x}_l)$, BT assumes the following distribution:

$$\mathrm{P}^*_{\text{triplet}}(\mathbf{x}_w \succ \mathbf{x}_l|\mathbf{x}_{\text{prompt}}) = \frac{\exp(r^*(\mathbf{x}_w, \mathbf{x}_{\text{prompt}}))}{\sum_{\mathbf{y}\in\{\mathbf{x}_l, \mathbf{x}_w\}} \exp(r^*(\mathbf{y}, \mathbf{x}_{\text{prompt}}))} \tag{1}$$

where r^* is the optimal latent reward model which generated the preferences and $\mathrm{P}^*_{\text{triplet}}$ is the optimal preference distribution. Through re-parameterization tricks (Rafailov et al., 2023) proposed to directly optimize the preference learning and bypassing the need to learn the reward explicitly. The final loss can be formulized as:

$$\mathcal{L}_{\mathrm{R}} = -\mathbb{E}_{(\mathbf{x}_{\text{prompt}}, \mathbf{x}_w, \mathbf{x}_l)\sim \mathrm{D}_{\text{triplet}}} \Bigg[\log \sigma \Bigg(\beta \log \Big(\frac{\mathrm{P}_\theta(\mathbf{x}_w|\mathbf{x}_{\text{prompt}})}{\mathrm{P}_{\text{pair}}(\mathbf{x}_w|\mathbf{x}_{\text{prompt}})} \Big) - \beta \log \Big(\frac{\mathrm{P}_\theta(\mathbf{x}_l|\mathbf{x}_{\text{prompt}})}{\mathrm{P}_{\text{pair}}(\mathbf{x}_l|\mathbf{x}_{\text{prompt}})} \Big) \Bigg) \Bigg] \tag{2}$$

where P_θ is the parameterized model to learn the preferences and P_{pair} is the model trained on pairs in previous stage for local editing and will be considered as fixed reference distribution.

3.5. Inference and evaluation

During inference, the model starts with an initial seed sequence, iteratively edits and is expected to improve its fit-

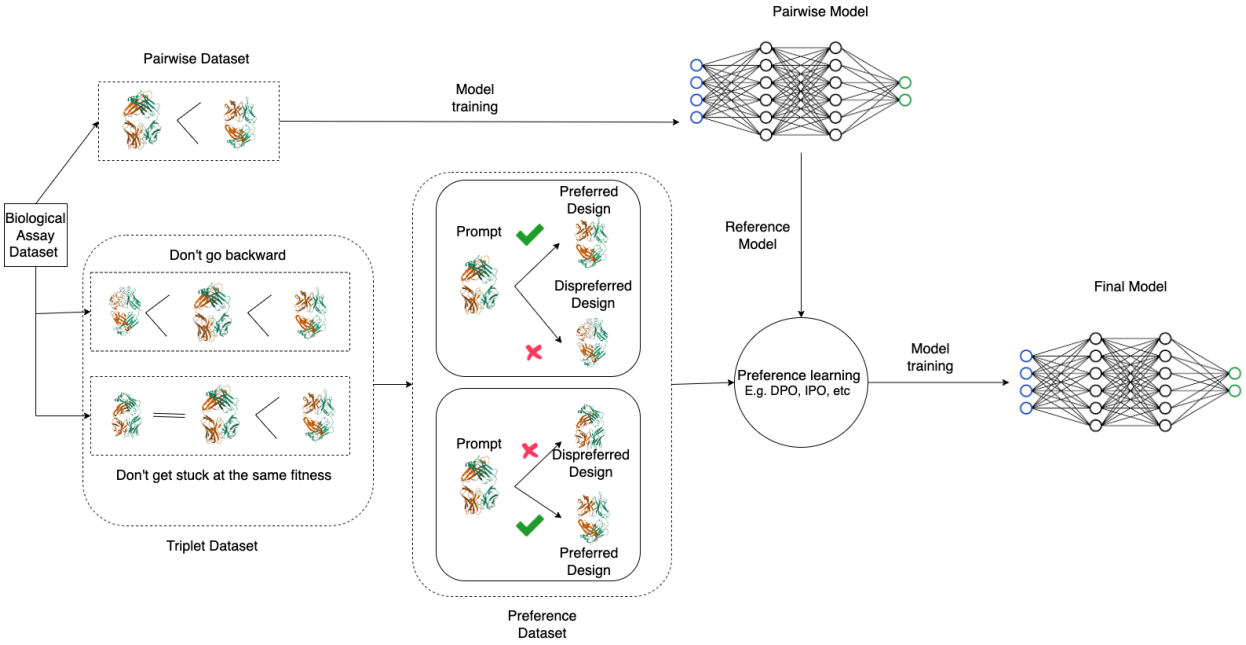

Figure 1. Schematic overview of extrapolative protein design through triplet preference learning.

ness. At iteration t given the seed sequence $\mathbf{x}_{t-1}$ and the trained extrapolative protein design model $\mathrm{P}_{\mathrm{triplet}}(.|\mathbf{x})$, one would sample $\mathbf{x}_t \sim \mathrm{P}_{\mathrm{triplet}}(.|\mathbf{x}_{t-1})$ until t reaches T (i.e. 10) predefined iterations. In practice, we start with set of initial sequences with their fitness very close to wild-type. For each initial seed sequence, we sample N (i.e. 10 for AAV and 2 for GFP) sequences using combination of top-k and top-p sampling with $k = 10$, $p = 0.95$ and a temperature of 0.7. At the end of each iteration, we randomly select K (i.e. 10,000 for AAV and 2,000 for GFP) samples from all generated sequences and use them as seeds for next iteration. At the last Tth iteration, we evaluate the final K samples. For *in-silico* evaluation of GFP and AAV datasets, we used evaluators trained by (Kirjner et al., 2023).

4. Experiments

4.1. Datasets

In order to assess the extrapolation ability of models on both sequence and fitness landscape, we have utilized the Adeno-associated virus (AAV) and Aequorea victoria GFP (avGFP) datasets processed by Kirjner et al. (2023). They proposed *mutational gap* as minimum number of mutations required from the training set to achieve the optimal fitness as a way to measure extrapolation ability of protein design models. We used the medium difficulty datasets with mutational gap of 6 between any sequence in the training set to any high-fitness sequence in the 99th percentile. The *training region* for GFP dataset maps to a fitness range of $[1.31, 3.04]$ and the *extrapolation region* maps to fitness values exceeding > 3.04. The *training region* and *extrapolation region* for AAV dataset refer to fitness in the range of $[0, 7]$ and > 7 respectively.

4.2. Benchmarked models

We compared our proposed method to (i) *Sampling*: unconditional protein design through Prot-T5-XL (Elnaggar et al., 2021) (ii) *Iterative Controlled Extrapolation (ICE)*: extrapolation through learning a local editor by translating proteins with lower fitness to slightly better fitness (Padmakumar et al., 2023) (iii) *Align-plm*: extrapolation via Bradley-Terry (BT) model of ranked proteins with big enough distances (Lee et al., 2023). We could not compare our method against Genhance (Chan et al., 2021) as we couldn't run their code. For preference learning models, we focused only on the DPO model (Rafailov et al., 2023).

4.3. Implementation details

We used the CNN models trained by (Kirjner et al., 2023) and (Dallago et al., 2021) on smoothed fitness landscape of *training regions* of GFP and AAV datasets respectively and utilized them as scorer functions f_s. Following (Padmakumar et al., 2023), we created the pairs dataset $\mathrm{D}_{\mathrm{pairs}} = \{(\mathbf{x}_1^i, \mathbf{x}_2^i)\}_{i=1}^{M}$ with $M = 900K(100K)$ training (validation) samples where they follow $|f_s(\mathbf{x}_1) - f_s(\mathbf{x}_2)| < 0.5$. We trained the local editor model on $\mathrm{D}_{\mathrm{pairs}}$ for 10 epochs with the AdamW optimizer (Loshchilov & Hutter, 2017), a learning rate of 1e-4 and batch size of 384.

Next, we created the preference dataset for both proteins following the principles of (i) *Don't go backward* and (ii) *Don't get stuck at the same fitness*. For GFP, we binned sequences based on their fitness to buckets of $[0, 0.25, 0.75, 1, 1.25, 1.5, 1.75, 2, 2.25]$ (smooth fitness range is different from the actual fitness measured from wet lab). For AAV, we binned sequences based on their fitness buckets of $[-100, -6, -5.5, -5, -4.5, -4, -3.5, -3, -2.5, -2, -1.5, -1$ In total, we created 100K and 10K training and validation samples respectively. We further fine-tuned the local editor model based on triplet-based preference learning through DPO for 1 epoch with batch size of 32, learning rate of 5e-7, $\beta = 0.1$ and the AdamW optimizer (Loshchilov & Hutter, 2017).

4.4. Results

Figure 2 shows that for both GFP (top) and AAV (bottom) datasets, the triplet-based preference learning approach outperforms baseline models in generating sequences with fitness in the extrapolation region. The *align-plm* model has higher mean fitness in comparison to *ICE* model, but its top 100 generated sequences perform worse than *ICE*. The top 100 generated sequences from the DPO model for GFP have an average fitness of 3.66, compared to ICE and align-plm, which have averages of 2.39 and 2.12, respectively. Similarly, for the AAV dataset, the top 100 sequences generated by the DPO model have an average fitness of 11.13, whereas ICE and align-plm have averages of 9.50 and 8.70, respectively.

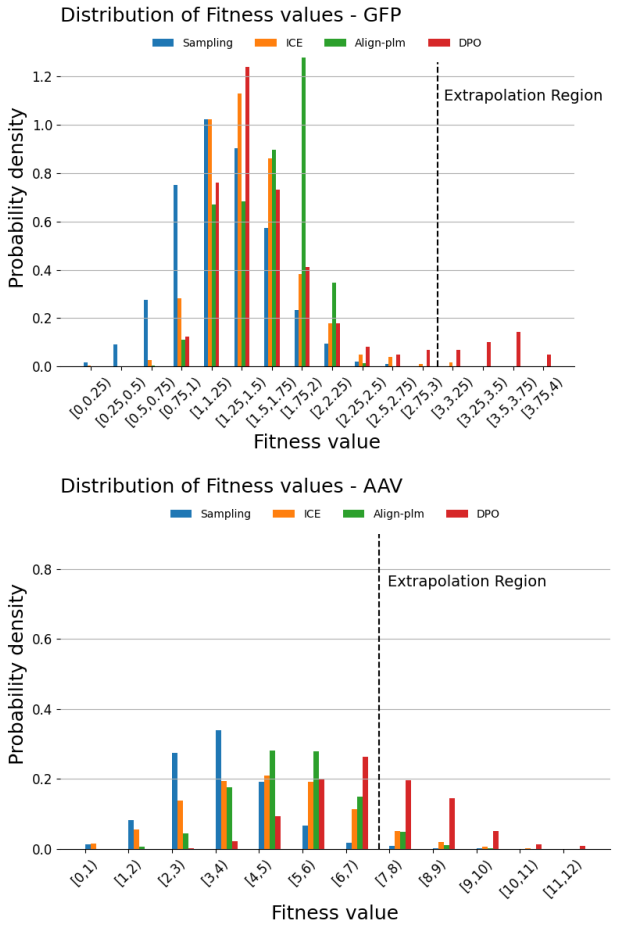

Figure 2. Comparison of in-silico fitness evaluation for baselines and proposed method (top) GFP dataset (bottom) AAV dataset.

5. Mutational analysis

We analyzed the sequences generated by triplet-based preference learning (DPO) in comparison to baselines. We utilized the ProstT5 model (Heinzinger et al., 2023) trained in multimodal fashion (sequence and structure) to embed unique sequences generated by each method. Two dimensional visualization of embeddings through t-SNE (Van der Maaten & Hinton, 2008) in Figure 3 highlights that triplet-based preference learning rejects several regions with low fitness in embedding space and focuses more on specific regions with higher fitness, especially in the extrapolation region. Similarly, as shown in Figure 19, the sequences generated by the DPO and ICE models are located in different parts of the embedding space compared to align-plm. Logo plots of generated sequences for baselines and DPO are shown in Figures 20 for AAV and 20 for GFP. We further showed that (i) triplet-based preference learning (DPO) magnifies the mutation rate of several residues and pushes the rest of them to lower values compared to ICE (Figures 21 for AAV and 10 for GFP) and (ii) the differences between the log probability of desired (chosen) sequences vs undesired (rejected) ones and accuracy of correctly distinguishing them increase for the validation set (Figures 3 and 17 respectively), which indicates that the model would be guarded against generating undesired sequences dramatically as expected.

6. Ablation studies

6.1. Effect of scorer

We assessed the effect of scorer in inference for baselines (ICE and Prot-T5-XL) and preference learning-based extrapolative models. Based on distribution of fitness shown in Figure 4 for AAV dataset, preference learning with triplets outperforms baselines with a large margin. As expected, scorers can enhance the performance of extrapolative models in comparison to versions without scorers (ICE + scorer vs ICE 15, DPO + scorer vs DPO 16).

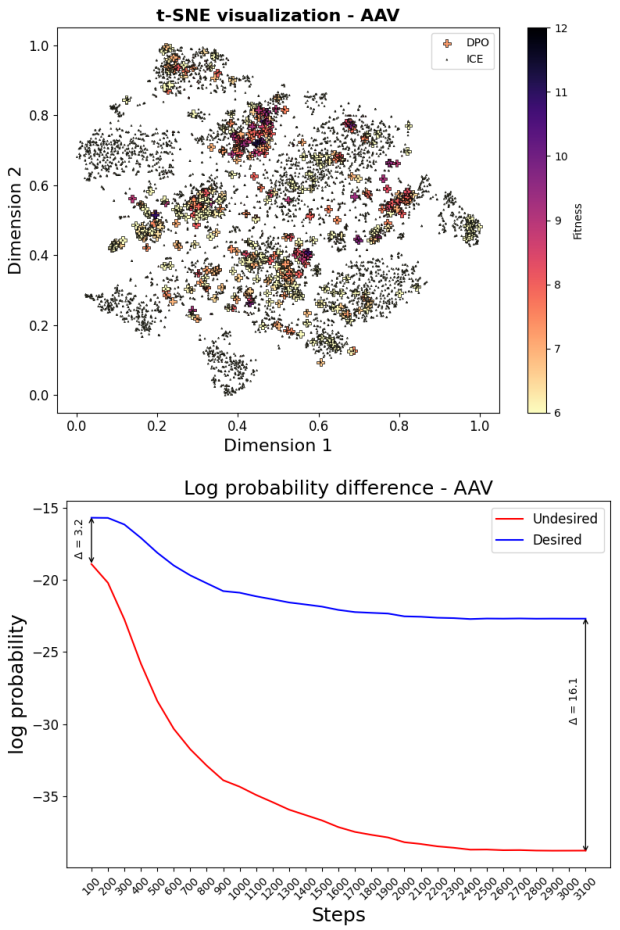

Figure 3. (top) t-SNE visualization (2 dimensions) of generated sequences for ICE vs DPO based on ProstT5 embedding (Heinzinger et al., 2023) (bottom) Log probability of validation set's desired vs undesired sequences for DPO model.

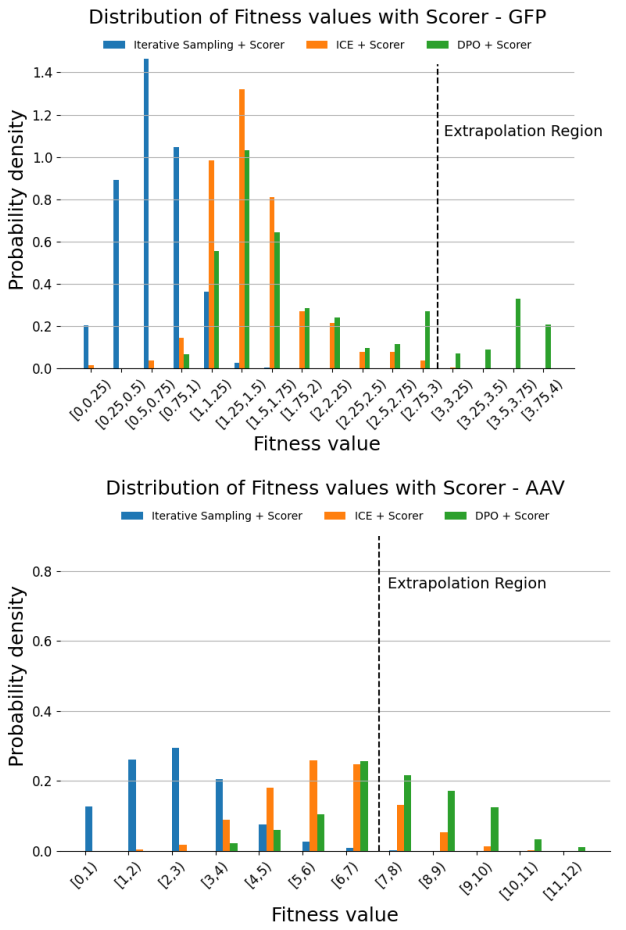

Figure 4. Comparison of in-silico fitness evaluation of generated sequences for baselines and proposed method with scorer (top) GFP dataset (bottom) AAV dataset.

6.2. Number of iterations

We assess the effect of number of iterations on both ICE model and our proposed triplet-based preference learning approach (DPO). Based on the results presented in Figures 22 and 14 for the AAV and GFP datasets, we observed that triplet-based preference learning actually benefits considerably from 10 iterations. For the triplet-based preference learning model on the AAV dataset, in the first iteration it had an average fitness of 5.11 and in the fifth and tenth iterations the average fitness moved to 6.10 and 6.73 respectively. Additionally, the average fitness of the top 100 candidates proposed in the fifth and tenth iterations were 10.69 and 11.13 respectively.

7. Conclusion

We presented a triplet-based preference learning framework for extrapolative protein design. Our framework significantly outperforms baseline models that learn only from pairwise relationships on designing sequences with higher fitness for both AAV and GFP datasets where the model needed to extrapolate both on sequence and fitness spaces. Potential future directions include (1) assessing the effect of higher order relationship (quadruples etc.) through Plackett-Luce ranking models (Plackett, 1975; Luce, 2005) on extrapolation, (2) benchmarking of recently proposed preference learning methods such as Kahneman-Tversky Optimization (KTO) (Ethayarajh et al., 2024) and Identity-mapping preference optimization (IPO) (Azar et al., 2023) against DPO, (3) utilizing reasoning approaches such as tree of thoughts (Yao et al., 2024) to boost performance of the proposed extrapolative protein design model.

References

Azar, M. G., Rowland, M., Piot, B., Guo, D., Calandriello, D., Valko, M., and Munos, R. A general theoretical paradigm to understand learning from human preferences. *arXiv preprint arXiv:2310.12036*, 2023.

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

Chan, A., Madani, A., Krause, B., and Naik, N. Deep extrapolation for attribute-enhanced generation. *Advances in Neural Information Processing Systems*, 34:14084–14096, 2021.

Christiano, P. F., Leike, J., Brown, T., Martic, M., Legg, S., and Amodei, D. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017.

Dallago, C., Mou, J., Johnston, K. E., Wittmann, B. J., Bhattacharya, N., Goldman, S., Madani, A., and Yang, K. K. Flip: Benchmark tasks in fitness landscape inference for proteins. *bioRxiv*, pp. 2021–11, 2021.

Elnaggar, A., Heinzinger, M., Dallago, C., Rehawi, G., Wang, Y., Jones, L., Gibbs, T., Feher, T., Angerer, C., Steinegger, M., 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.

Ethayarajh, K., Xu, W., Muennighoff, N., Jurafsky, D., and Kiela, D. Kto: Model alignment as prospect theoretic optimization. *arXiv preprint arXiv:2402.01306*, 2024.

Heinzinger, M., Weissenow, K., Sanchez, J. G., Henkel, A., Steinegger, M., and Rost, B. Prostt5: Bilingual language model for protein sequence and structure. *bioRxiv*, pp. 2023–07, 2023.

Kirjner, A., Yim, J., Samusevich, R., Bracha, S., Jaakkola, T. S., Barzilay, R., and Fiete, I. R. Improving protein optimization with smoothed fitness landscapes. In *The Twelfth International Conference on Learning Representations*, 2023.

Kreutzer, J., Uyheng, J., and Riezler, S. Reliability and learnability of human bandit feedback for sequence-to-sequence reinforcement learning. *arXiv preprint arXiv:1805.10627*, 2018.

Lee, M., Lee, K., and Shin, J. Fine-tuning protein language models by ranking protein fitness. In *NeurIPS 2023 Generative AI and Biology (GenBio) Workshop*, 2023.

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Luce, R. D. *Individual choice behavior: A theoretical analysis*. Courier Corporation, 2005.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

Padmakumar, V., Pang, R. Y., He, H., and Parikh, A. P. Extrapolative controlled sequence generation via iterative refinement. In *International Conference on Machine Learning*, pp. 26792–26808. PMLR, 2023.

Peng, B., Song, L., Tian, Y., Jin, L., Mi, H., and Yu, D. Stabilizing rlhf through advantage model and selective rehearsal. *arXiv preprint arXiv:2309.10202*, 2023.

Plackett, R. L. The analysis of permutations. *Journal of the Royal Statistical Society Series C: Applied Statistics*, 24 (2):193–202, 1975.

Rafailov, R., Sharma, A., Mitchell, E., Manning, C. D., Ermon, S., and Finn, C. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36, 2023.

Van der Maaten, L. and Hinton, G. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

Xu, K., Zhang, M., Li, J., Du, S. S., Kawarabayashi, K.-i., and Jegelka, S. How neural networks extrapolate: From feedforward to graph neural networks. *arXiv preprint arXiv:2009.11848*, 2020.

Yao, S., Yu, D., Zhao, J., Shafran, I., Griffiths, T., Cao, Y., and Narasimhan, K. Tree of thoughts: Deliberate problem solving with large language models. *Advances in Neural Information Processing Systems*, 36, 2024.

A. GFP experiments

A.1. Effect of Scorer

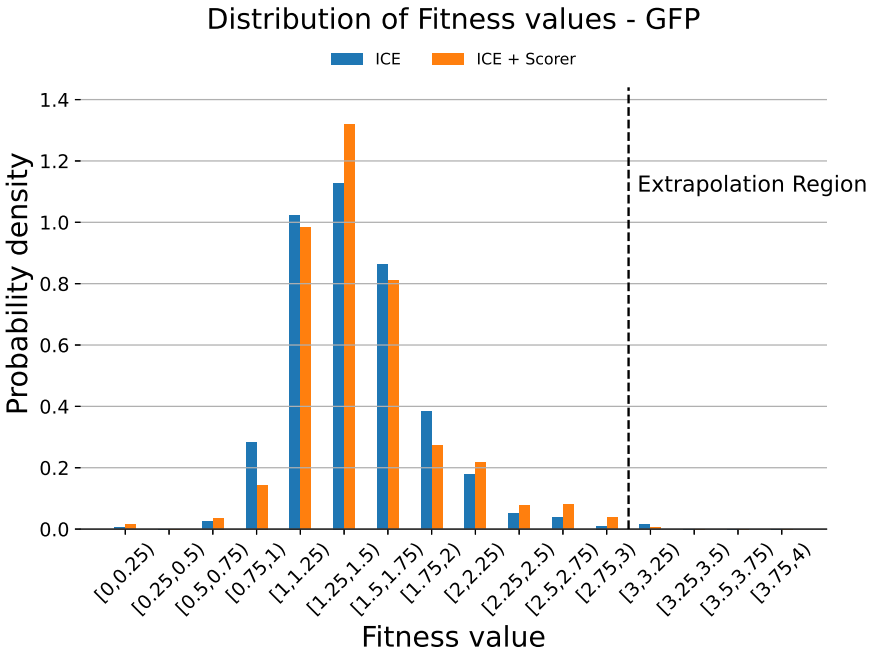

Figure 5. Comparison of in-silico fitness evaluation of generated sequences for ICE vs ICE with scorer.

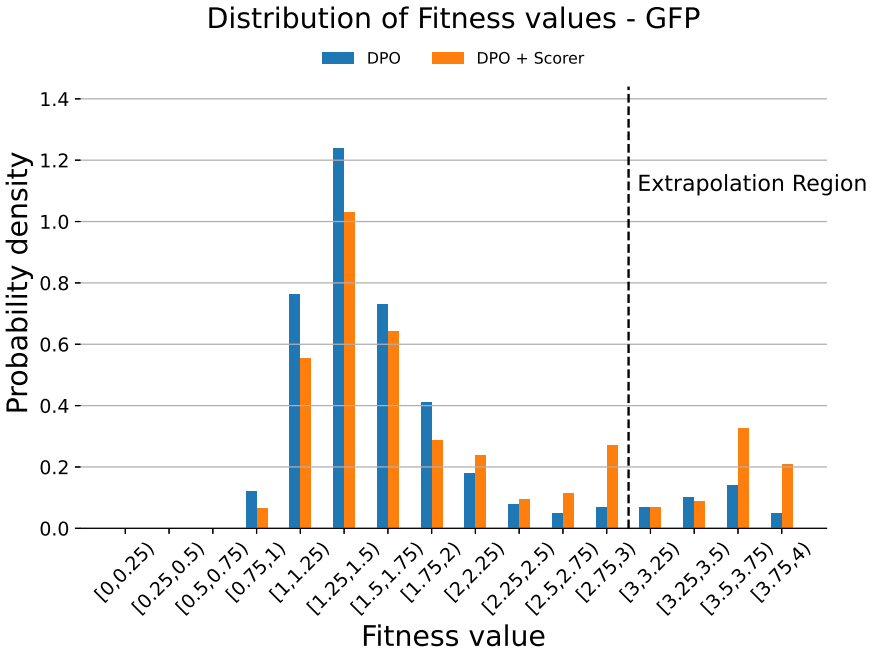

Figure 6. Comparison of in-silico fitness evaluation of generated sequences for DPO vs DPO with scorer.

A.2. Mutational analysis

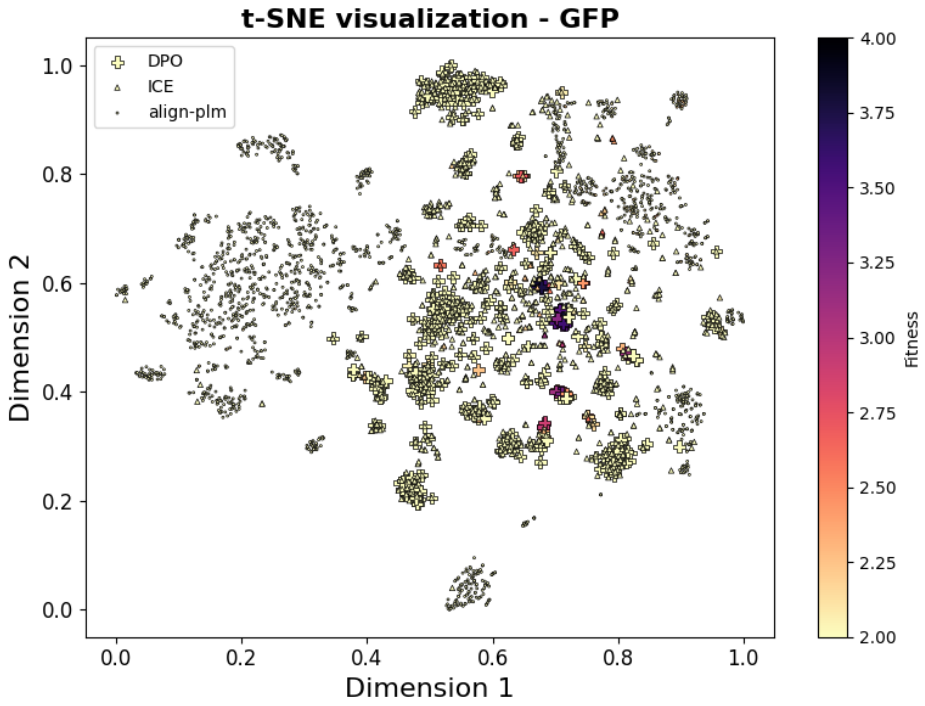

Figure 7. t-SNE visualization of generated sequences for different models based on ProstT5 embedding (Heinzinger et al., 2023).

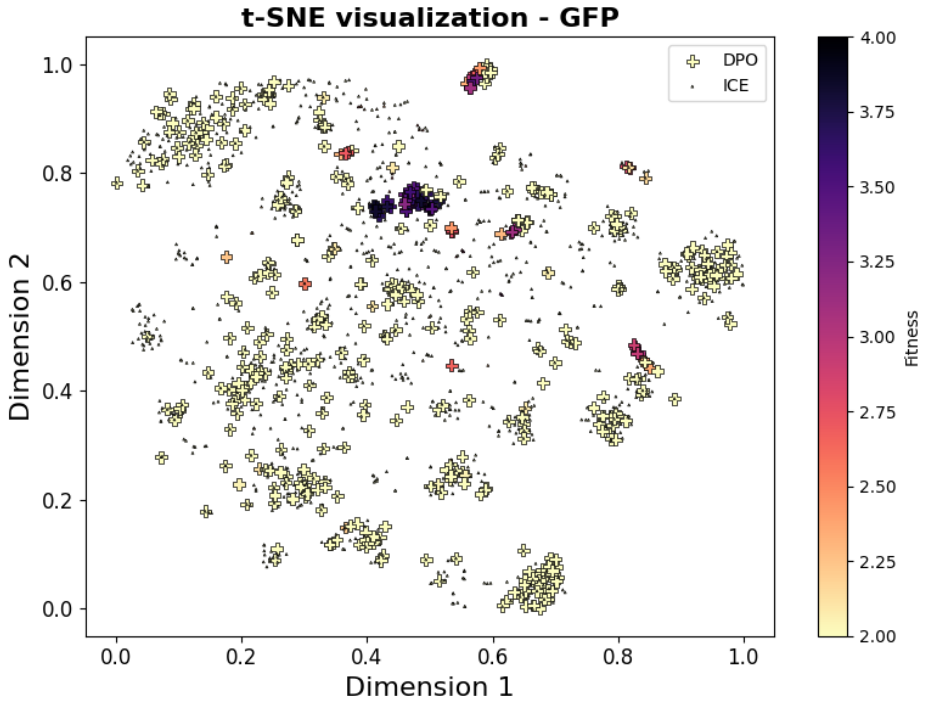

Figure 8. t-SNE visualization of generated sequences for ICE vs DPO based on ProstT5 embedding (Heinzinger et al., 2023).

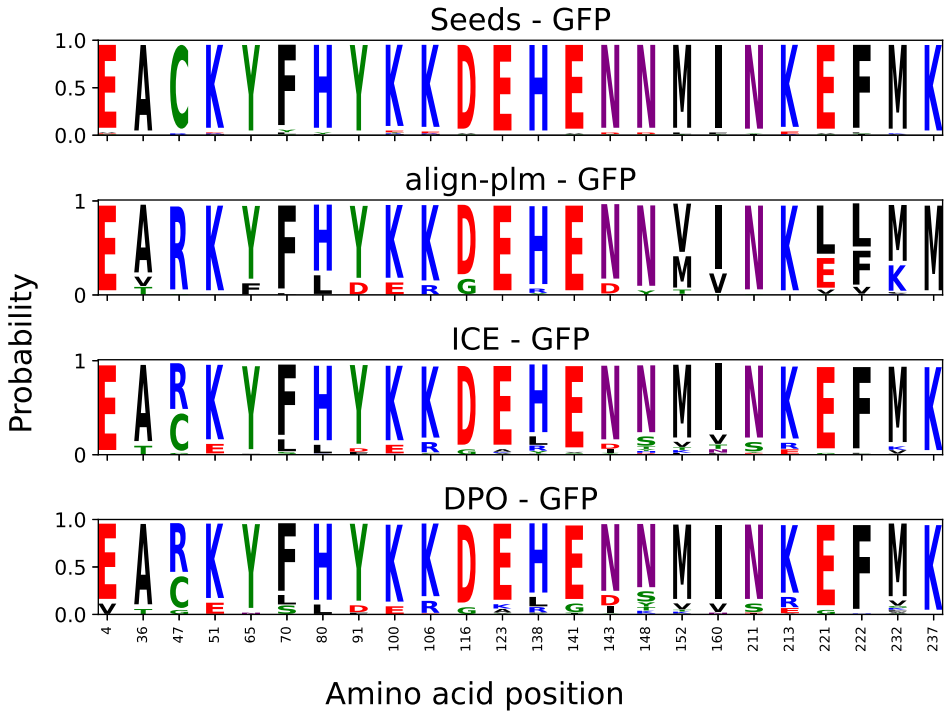

Figure 9. Amino acid logo plot of generated sequences from various extrapolative models (For positions with at least 10% mutation in population of any method).

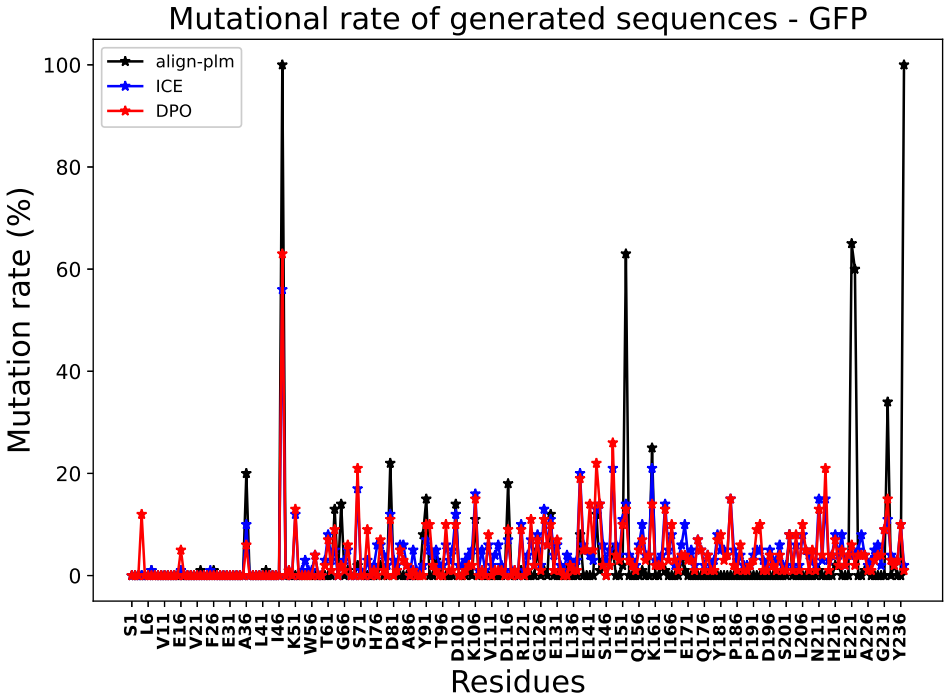

Figure 10. Mutational rate of generated sequences from DPO, align-plm and ICE.

A.3. Accuracy of predicting desired vs undesired sequences

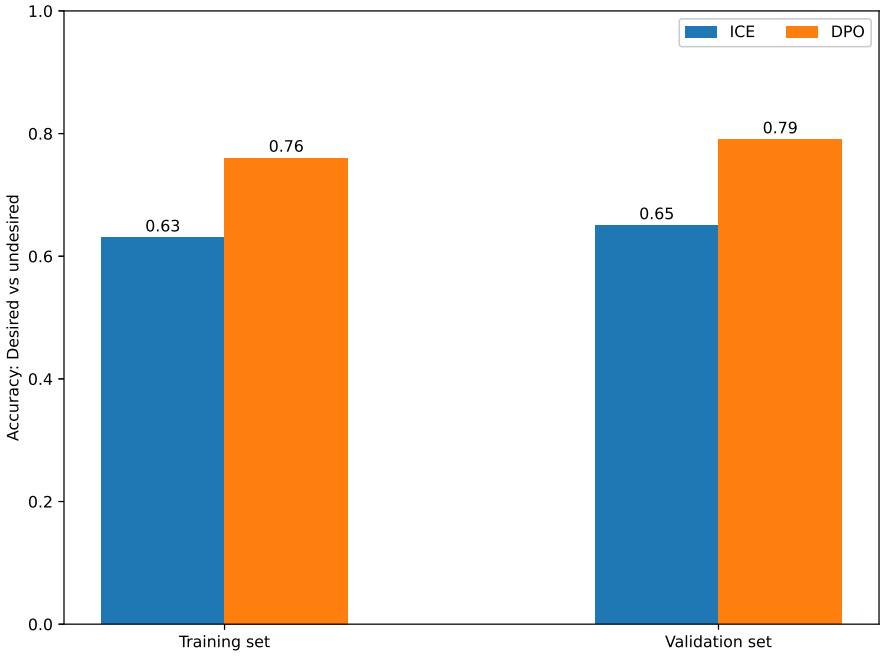

Figure 11. Comparison of distinguishing desired vs undesired pairs for DPO vs ICE model.

A.4. Log probability of desired vs undesired sequences

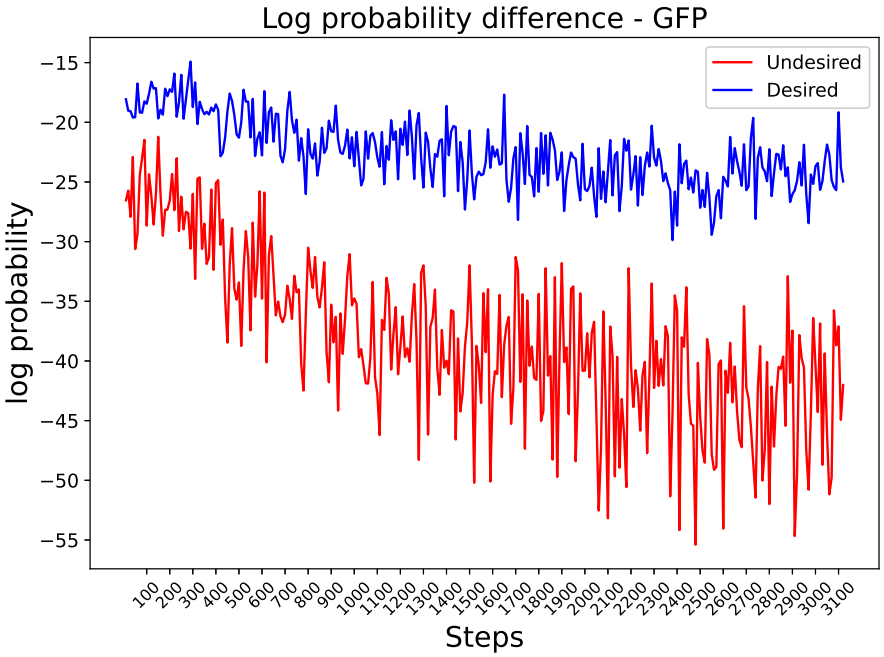

Figure 12. Log probability of training set's desired vs undesired sequences for DPO model.

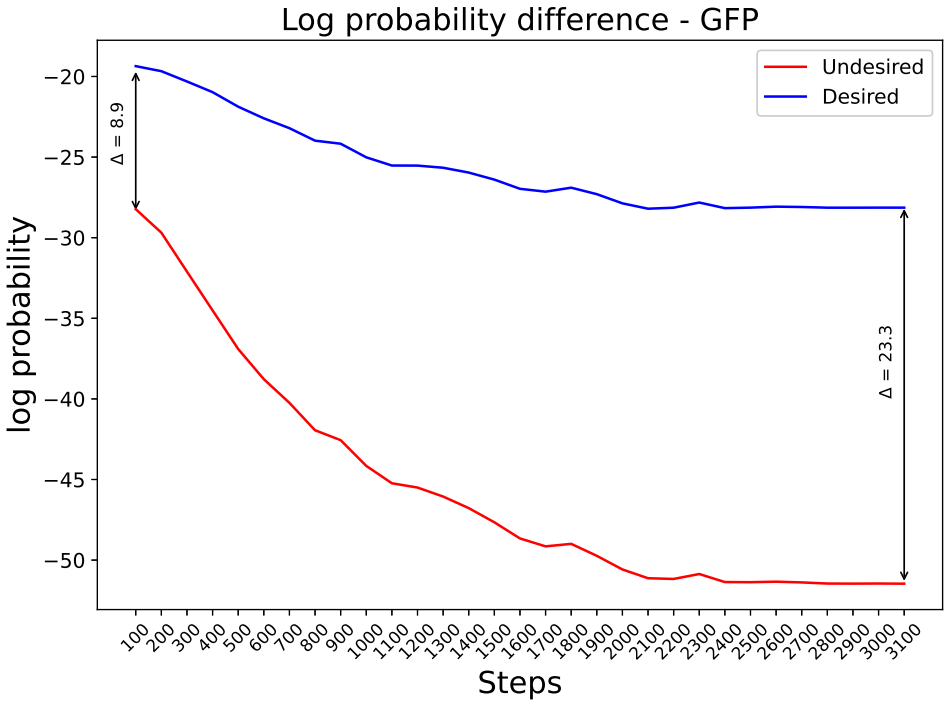

Figure 13. Log probability of validation set's desired vs undesired sequences for DPO model.

A.5. Number of Iterations

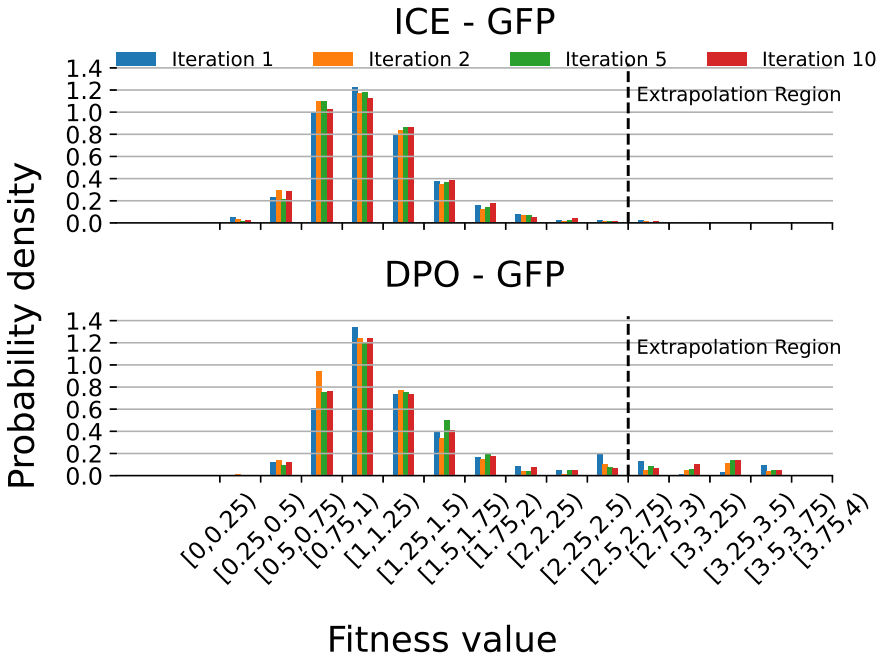

Figure 14. Comparison of in-silico fitness evaluation of generated sequences for ICE (top) and DPO (bottom) for different iterations.

B. AAV experiments

B.1. Effect of scorer

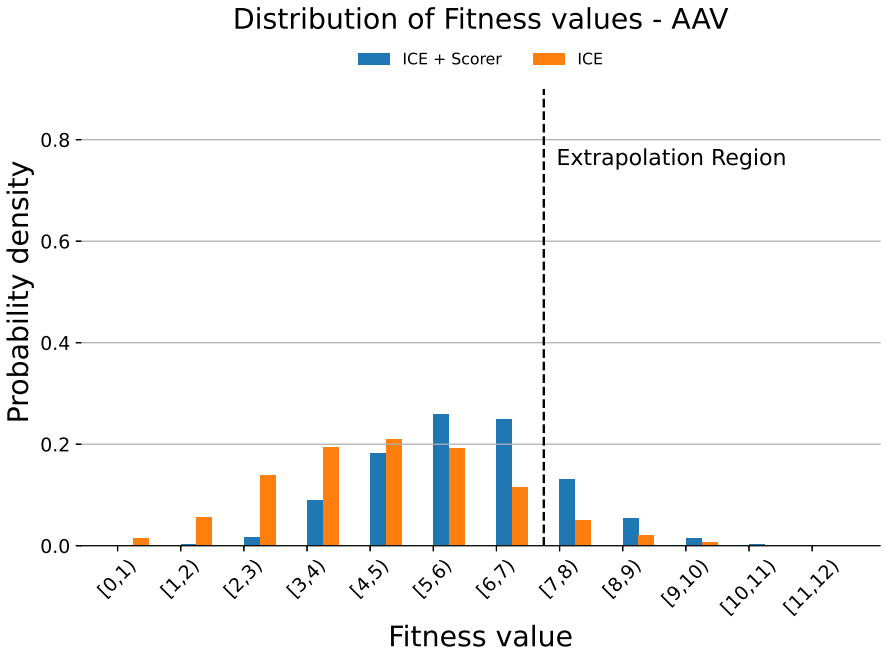

Figure 15. Comparison of in-silico fitness evaluation of generated sequences for ICE vs ICE with scorer.

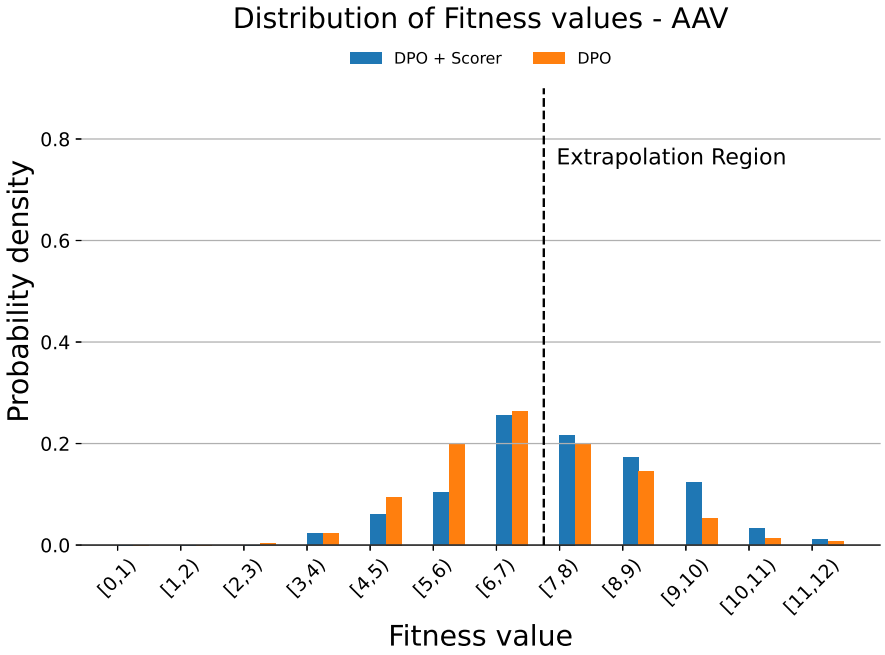

Figure 16. Comparison of in-silico fitness evaluation of generated sequences for DPO vs DPO with scorer.

B.2. Accuracy of predicting desired vs undesired sequences

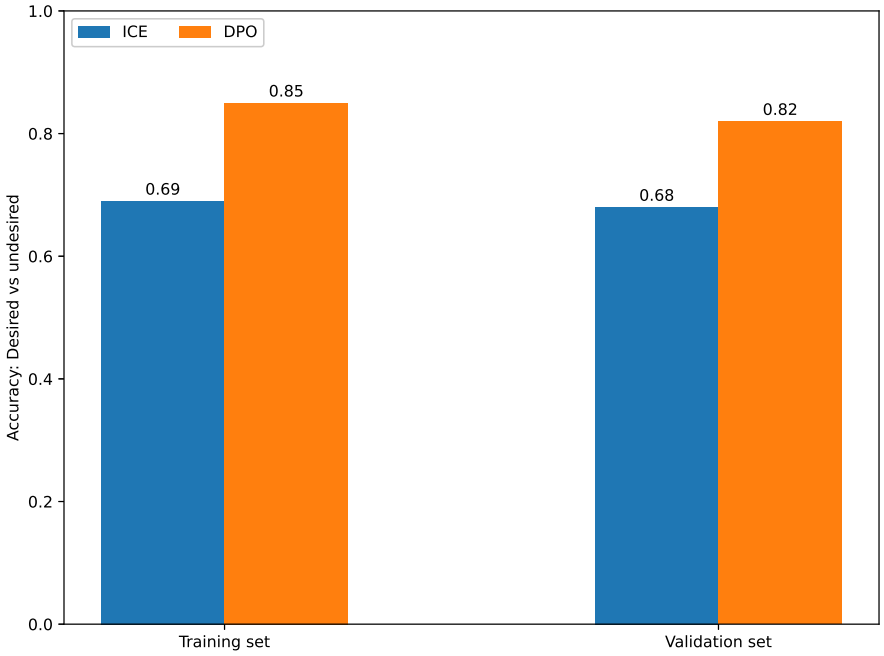

Figure 17. Comparison of distinguishing desired vs undesired pairs for DPO vs ICE model.

B.3. Log probability of desired vs undesired sequences

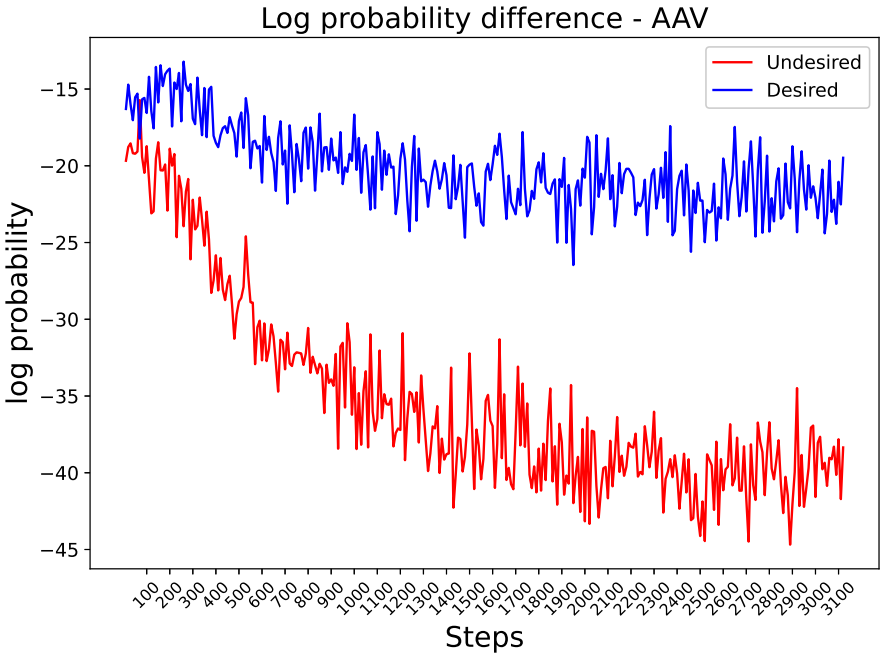

Figure 18. Log probability of training set's desired vs undesired sequences for DPO model.

B.4. Mutational analysis

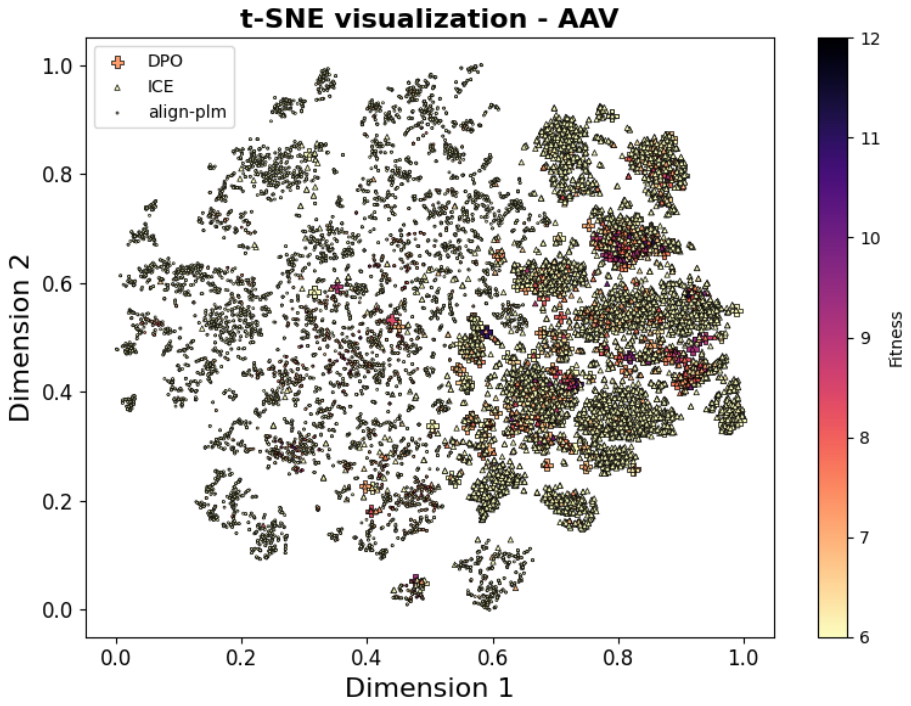

Figure 19. t-SNE visualization of generated sequences for different models based on ProstT5 embedding (Heinzinger et al., 2023).

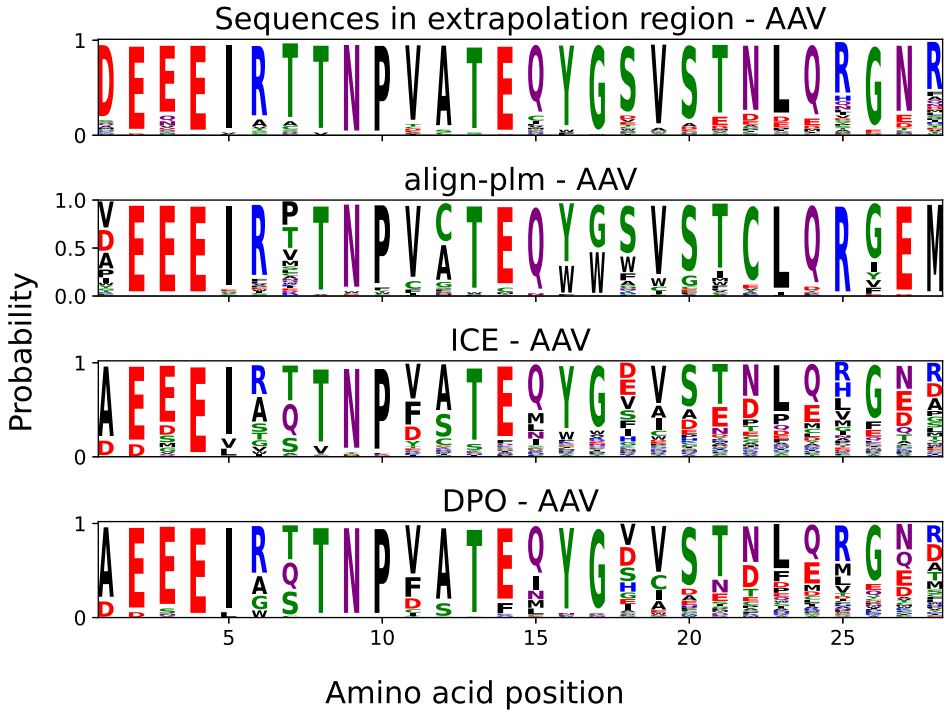

Figure 20. Amino acid logo plot of generated sequences from various extrapolative models.

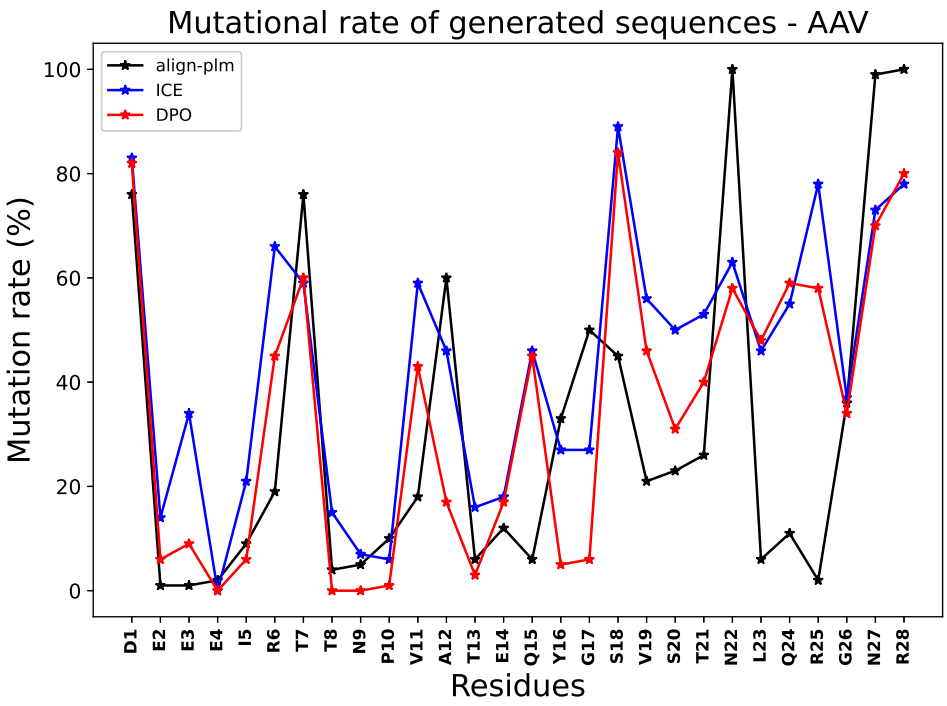

Figure 21. Mutational rate of generated sequences from DPO, align-plm and ICE.

B.5. Number of iterations

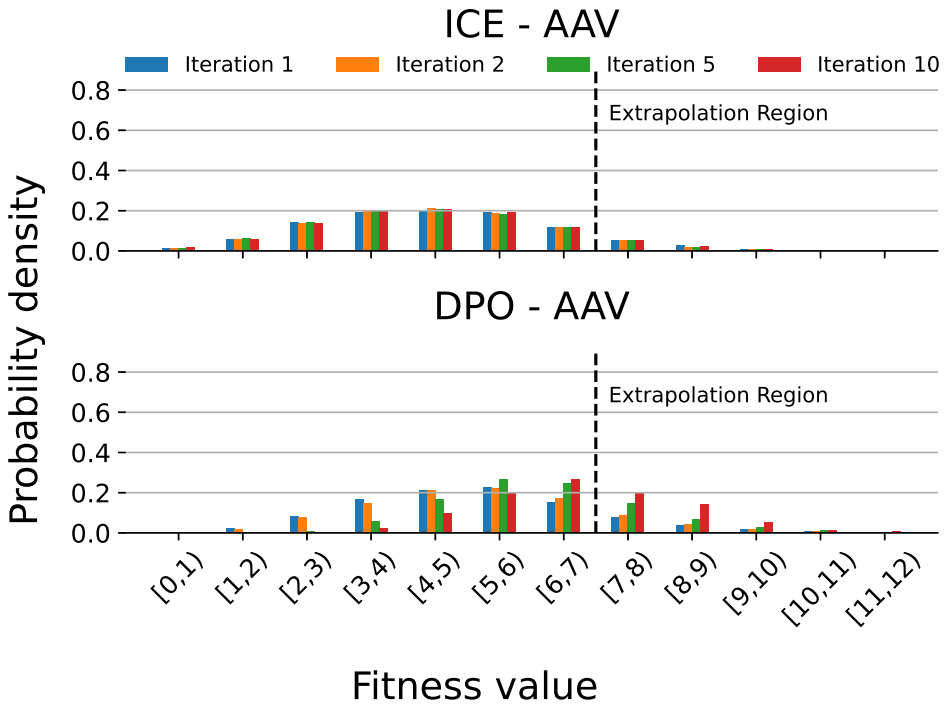

Figure 22. Comparison of in-silico fitness evaluation of generated sequences for ICE (top) and DPO (bottom) for different iterations.