Learning New Biophysical Controls in Protein Language Models via Supervised and Preference-Based Fine-Tuning

Nahum Maru

June 10, 2025

Abstract

Protein language models (pLMs) have shown strong capabilities in modeling protein structure and function, but their controllability remains limited to coarse functional tags like Enzyme Commission (EC) numbers. In this work, we introduce a framework to extend pLM controllability to include biophysical properties, focusing on protein folding stability—a key factor in protein design. We build on ZymCtrl, a prompt-based conditional model, by introducing new control tags such as `stability="high"` and fine-tune it through a hybrid method. First, we apply supervised fine-tuning (SFT) on sequences labeled with stability scores from Rosetta energy predictions. Next, we use Direct Preference Optimization (DPO) with pairwise preferences derived from either random sampling or BLOSUM-guided mutations, where more stable sequences are preferred. Our approach is notable for combining explicit prompting with preference-based fine-tuning to target stability as a controllable generation property.

Our results show that SFT successfully enables coarse but reliable control over sequence stability. Prompted generations shift appropriately in predicted ΔG values, and structural viability (measured by pLDDT from ESMFold) remains high. However, DPO proves fragile: training on roughly 1,000 preference pairs leads to unstable learning dynamics and eventual collapse in generation quality. Attempts to stabilize DPO using different pair construction strategies or model initializations were unsuccessful.

These findings suggest that SFT is a robust method for introducing new biophysical controls, but DPO requires stronger supervision and better-curated data to be effective in this domain. Future work will explore other properties such as solubility, incorporate experimental evolutionary data for more aligned preferences, and investigate curriculum learning to stabilize DPO optimization. Our framework represents a step toward more flexible, property-aware generation in protein language models.

1 Introduction

Protein language models (pLMs) have demonstrated remarkable capabilities in modeling protein structure, function, and fitness. Models such as ESM-1b and ZymCtrl leverage transformer-based architectures to learn rich representations of amino acid sequences. However, the ability to control generation remains limited, typically restricted to functional tags like Enzyme Commission (EC) numbers.

In this work, we introduce a framework for fine-grained control over protein folding stability—an essential biophysical property in protein engineering. Building on ZymCtrl, we incorporate new textual control tags (e.g., `stability="high"`) and develop a hybrid training pipeline that combines supervised fine-tuning (SFT) and Direct Preference Optimization (DPO). Our approach first introduces the control tags via SFT and then refines generation using pairwise preference data derived from predicted thermodynamic stability.

We find that supervised fine-tuning (SFT) enables reliable, though coarse-grained, control over protein stability via newly introduced textual prompts. In contrast, Direct Preference Optimization (DPO) exhibits significantly less stability: training is fragile, and gains in controllability are inconsistent. Even with carefully constructed preference data, DPO often results in degraded sequence quality and diminished sensitivity to prompts. These findings suggest that while SFT offers a solid foundation for extending controllability, unlocking the full potential of preference-based fine-tuning will require more precise and higher-quality pairwise supervision. We make our code available on GitHub.

2 Background and Related Work

Protein language models (pLMs) apply deep learning architectures—originally developed for natural language processing—to large corpora of amino acid sequences, capturing statistical patterns that relate to protein structure, function, and fitness Rives et al. [2021]. Early successes demonstrated that pretraining on millions of sequences enables strong performance on downstream tasks such as mutation effect prediction and contact inference.

ESM-1b Rives et al. [2021] set a new benchmark with a transformer-based model trained via masked language modeling, showing impressive zero-shot capabilities across a wide range of protein tasks. However, its generative outputs were untargeted, lacking mechanisms to control for specific functional or biophysical properties.

To address controllability, **ZymCtrl** Munsamy et al. [2024] introduced prompt-based conditional generation, enabling protein design conditioned on enzymatic function by prepending Enzyme Commission (EC) numbers as control tags. This allowed explicit steering of generations toward desired enzyme classes. However, ZymCtrl's conditioning is limited to EC numbers—effectively controlling for function but not other important properties such as folding stability or solubility.

Expanding the conditional modeling paradigm, **ProCALM** Yang et al. [2024] advanced the idea of prompt-based generation by integrating a broader range of natural language descriptors, including taxonomy labels, using supervised fine-tuning (SFT). Notably, ProCALM employed adapter-based fine-tuning to maintain the base model's general capabilities while enabling task-specific control, highlighting the importance of parameter-efficient methods.

To move beyond supervised learning, **ProteinDPO** Widatalla et al. [2024] adapted Direct Preference Optimization (DPO) for protein sequence generation. Rather than relying on labeled datasets, ProteinDPO fine-tunes models using pairwise preference data—allowing the model to align its outputs with biophysical properties like stability, based on experimental fitness data or proxy scores. This introduced a flexible route to optimizing sequences even in cases where explicit labels are scarce.

Taken together, these works illustrate the trajectory from unsupervised pretraining toward increasingly precise and controllable protein generation. Yet, most conditioning efforts have focused on well-defined functional classes (e.g., EC numbers), with less emphasis on fine-grained biophysical control. Moreover, while preference-based methods like ProteinDPO have demonstrated promise, they have not been combined with prompt-based conditioning in a way that extends controllability to new property spaces.

Our work builds on this foundation by targeting protein folding stability as a controllable property, combining prompt-based SFT with DPO for refinement. Crucially, the framework we propose is designed to be extendable to other biophysical properties, laying the groundwork for versatile, multi-objective control in protein generation tasks.

3 Our Approach

3.1 Controlling Protein Stability via Prompting

We extend the ZymCtrl model, which achieves functional conditioning by prepending control tags (e.g., `ec="4.2.1.1"`) to protein sequences. Our approach focuses specifically on protein folding stability, introducing new tags of the form `stability="high"`, `stability="medium"`, and `stability="low"`. These are implemented as simple textual prompts, requiring no modifications to the model architecture. We adopt a transformer-based pLM architecture similar to ESM-1b, retaining its tokenizer and positional encoding to ensure compatibility with pre-trained weights.

To train the model to recognize and respond to these new tags, we fine-tune it using supervised data labeled for stability. We repurpose the BRENDA enzyme dataset (which ZymCTRL uses for training) by annotating each sequence with rosetta energy scores, estimates thermodynamic stability (ΔG) for each protein. The continuous ΔG values are bucketed into three discrete stability classes, high, medium, and low, and the corresponding tag is prepended to each sequence. We then fine-tune the model with the standard autoregressive objective, enabling it to associate stability tags with meaningful sequence patterns.

3.2 Preference-Based Optimization with DPO

Direct Preference Optimization (DPO) Rafailov et al. [2023] is a fine-tuning method that steers language models using pairwise preferences, avoiding the need for explicit reward models or reinforcement learning algorithms like PPO. Instead of maximizing expected rewards, DPO trains the model to prefer responses ranked higher in a pairwise comparison. This is well-suited to protein design, where proxy metrics (e.g., predicted stability) can serve as useful indicators of sequence quality.

Formally, given preference pairs (x, y^+, y^-)—where x is the conditioning input (e.g., a stability tag), y^+ is the preferred sequence, and y^- is less preferred—DPO optimizes:

$$\mathcal{L}_{\text{DPO}} = -\log \sigma \left(\beta \left(\log \pi_\theta(y^+ \mid x) - \log \pi_\theta(y^- \mid x)\right)\right) \tag{1}$$

where σ is the sigmoid function, π_θ is the model distribution, and β controls preference sharpness.

While SFT teaches the model to interpret stability tags, its precision may be limited due to noisy or weak supervision from proxy labels. To improve controllability, we apply DPO using preference pairs derived from Rosetta energy scores. We generate these pairs using two strategies:

- **Random Sampling:** Selecting BRENDA sequences with large stability differences (as predicted by Rosetta) and pairing them to reflect preference.
- **BLOSUM-Based Mutations:** Starting from a given BRENDA sequence, we randomly sample 10 amino acid positions and mutate each using transition probabilities defined by the BLOSUM62 substitution score matrix. The resulting mutated sequence is folded using ESMFold, and its predicted stability is assessed using Rosetta. If the predicted structure is confident (mean pLDDT > 0.7), we form a preference pair between the original and mutated sequence, with the more stable sequence (lower Rosetta energy) treated as preferred.

This combination of global and local perturbation strategies allows us to create both coarse and fine-grained preference data. By explicitly training the model to prefer more stable sequences, we aim to enhance control fidelity beyond what is achievable with SFT alone.

3.3 Training Pipeline

Our training pipeline consists of three stages:

1. **Pre-training:** Initialize from the ZymCtrl model.
2. **SFT Phase:** Fine-tune (SFT) on stability-tagged sequences derived from the BRENDA dataset.
3. **DPO Phase:** Construct preference pairs offline using Rosetta stability scores, and fine-tune the model using Direct Preference Optimization (DPO).

This hybrid strategy combines explicit supervision with offline preference-based fine-tuning, aiming to enable robust and precise control over protein stability. We construct preference pairs in advance using Rosetta scores, allowing the model to learn from curated examples of stability differences.

While our current work focuses exclusively on protein stability, the same methodology is extensible to other biophysical properties in future applications. This approach lays the foundation for multi-objective protein design, combining functional specificity with nuanced control over key biochemical attributes.

4 Experiments and Results

Our goal is to enable controllable generation of protein sequences conditioned on folding stability, while preserving the general performance of the base ZymCtrl model. This section details our evaluation metrics, data construction pipeline, and results from both supervised fine-tuning (SFT) and preference-based optimization using Direct Preference Optimization (DPO).

4.1 Evaluation Metrics

We evaluate models along two core dimensions:

- **Controllability:** To measure whether the model meaningfully responds to stability prompts, we generate sequences conditioned on each stability tag (`low`, `medium`, `high`) and analyze the distribution of their predicted folding stability scores (Rosetta total energy). A successful model should shift output distributions in the expected direction: lower energy for `<stability="low">` and higher for `<stability="high">`.
- **Model Fidelity:** To ensure that fine-tuning does not degrade general performance, we evaluate base model metrics including perplexity on EC-only prompts (i.e., without stability tags) and structure prediction confidence (pLDDT) from ESMFold. High pLDDT across generations indicates that the model continues to produce foldable proteins.

4.2 Data Labeling and Pair Construction

We use protein sequences from the BRENDA enzyme database, which ZymCtrl was originally trained on, and annotate them with Rosetta total energy scores as a proxy for folding stability—lower values correspond to more stable folds. We process these sequences for both SFT and DPO training as follows:

Supervised Fine-Tuning (SFT). To create labeled data for supervised learning, we discretize the Rosetta scores into three stability bins—`low`, `medium`, and `high`—based on dataset quantiles. Each sequence is prepended with a control tag indicating its class. We add these three new control tags to the tokenizer. For example:

```
<stability="low"> MKTFFVAGIL...
```

To ensure training quality, we filter out sequences with high perplexity or ESMFold confidence scores (pLDDT) below 0.7. The resulting dataset contains roughly 1,500 sequences per stability class, for a total of 4,500 labeled sequences.

Direct Preference Optimization (DPO). To construct pairwise preferences for DPO training, we generate pairs (x, y^+, y^-) where both sequences are conditioned on the same stability tag x, and y^+ is the more stable of the two. We use two strategies to construct these pairs:

- **Random Sampling:** We select sequence pairs from BRENDA with large differences in Rosetta scores, and assign preference based on which sequence is more stable.
- **BLOSUM-Based Mutations:** For a given BRENDA sequence, we randomly mutate 10 positions using substitutions guided by BLOSUM62. The mutated sequence is folded with ESMFold and rescored with Rosetta. If the mutant exhibits a substantial ΔG change, we form a preference pair with the original sequence.

This approach provides both global (divergent) and local (minimally perturbed) preference data to guide DPO optimization.

4.3 Supervised Fine-Tuning Results

We first assess whether supervised fine-tuning (SFT) enables the model to internalize and respond to protein stability control. ZymCtrl is a 768M parameter transformer-based protein language model trained on enzyme sequences with functional (EC number) control tags. We explore two key SFT experiments: (1) fine-tuning on stable sequences without explicit control tags, and (2) full stability-conditioning using discrete textual prompts.

Experiment 1: Stability Fine-Tuning Without Prompts. In our first experiment, we fine-tune the model on 12,000 sequences from the `stability="low"` class (i.e., most stable) without introducing any new control tokens or prepended tags. The goal is to assess whether the model can be nudged toward generating more stable sequences via data distribution alone, while preserving base model behavior on EC-based tasks.

We observe that, relative to the original ZymCtrl model, generations from the fine-tuned model yield a downward shift in Rosetta ΔG scores, indicating higher predicted folding stability. The median Rosetta ΔG shifts from 414.29 to **197.19**. The improvement is modest but consistent across random seeds. Importantly, EC-only prompts still yield viable sequences with high pLDDT (median > 0.7), and perplexity remains within 3% of the original model, suggesting minimal performance degradation. See Figure 1 and Figure 2 for plots showcasing these results

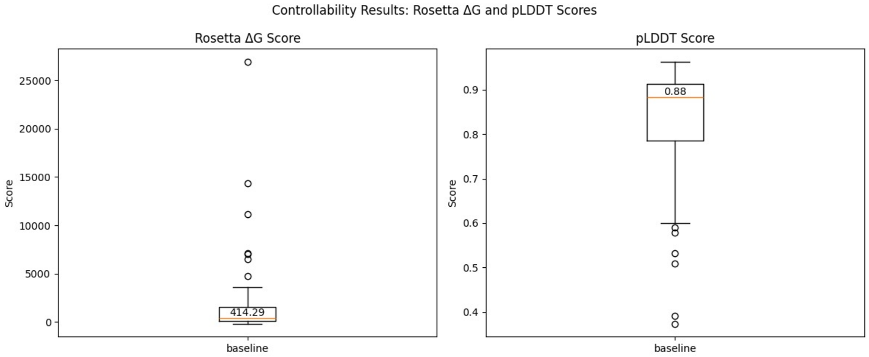

Figure 1: Distribution of predicted Rosetta ΔG scores for sequences generated by the pretrained ZymCtrl model using standard EC-number prompts. This serves as the baseline distribution for evaluating the effects of stability fine-tuning.

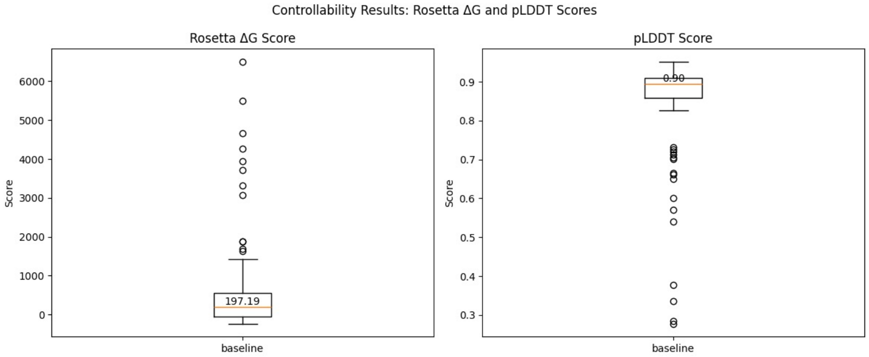

Figure 2: Distribution of predicted Rosetta ΔG scores for sequences generated by a model fine-tuned only on high-stability sequences from BRENDA, without using any control tags. Compared to the ZymCtrl baseline, the distribution shifts modestly toward more stable outputs, indicating that the model internalizes stability features even without explicit conditioning.

Experiment 2: Stability-Controlled Prompted SFT. In our main SFT experiment, we introduce three new control tokens—`<stability="low">`, `<stability="medium">`, and `<stability="high">`—into the tokenizer and fine-tune the model on 36,000 sequences (12,000 per class). Control tags are prepended in natural language format, and class labels are assigned via Rosetta ΔG score binning, as described earlier. We use full-parameter fine-tuning with the AdamW optimizer. We use a max learning rate of 5×10^{-5}, weight decay of 0.01, 100 warm up steps, cosine learning rate decay, and 4 training epochs.

We sample 200 sequences from each class and observe a clear shift in ΔG distributions: `<stability="low">` results in the most stable outputs, while `<stability="high">` shifts the distribution toward higher (less

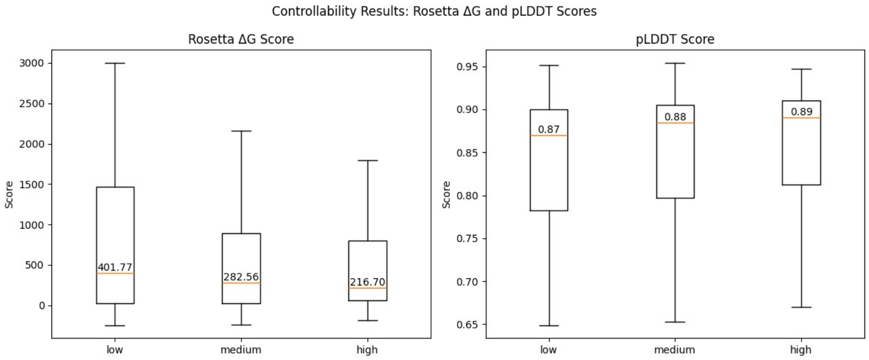

Figure 3: Distributions of predicted Rosetta ΔG for generations under each stability prompt after SFT. Lower scores indicate higher predicted stability.

stable) energy values (Figure 3). EC-number-only generations remain structurally viable (median pLDDT > 0.7), and perplexity is comparable to the base model, indicating that stability conditioning does not degrade general model performance.

Summary. Together, these experiments demonstrate that protein stability can be introduced as a controllable dimension via supervised fine-tuning. Even without explicit prompts, training on stable sequences biases the model toward generating more foldable proteins. Adding discrete textual prompts further enhances controllability while preserving the base model's functional capabilities.

4.4 Direct Preference Optimization Results

We next evaluate whether Direct Preference Optimization (DPO) can refine controllability beyond what SFT alone achieves. DPO fine-tunes the model on pairwise preferences between sequences with different predicted stability, encouraging higher likelihood on the more stable member of each pair. We fine-tune both the base ZymCtrl model and the SFT model using preference data constructed as described in Section 3.2, training for 3 epochs with AdamW and cosine learning rate decay. We experiment with $\beta \in \{0.1, 0.05\}$ to modulate preference sharpness.

Full Stability-Conditioned DPO. In our primary DPO experiment, we train the model using preference pairs drawn from across all stability classes, each pair prepended with an explicit stability tag. We observe that after one epoch of training, the model's generations remain nearly indistinguishable from the baseline—indicating no meaningful learning has occurred, as seen in Figure 4. However, continued training into the second and third epochs leads to rapid degradation: generated sequences become incoherent, excessively long, and largely unresponsive to prompts, as exhibited in Figure 5. This collapse in generation quality occurs consistently across both initialization settings (base and SFT models) and hyperparameters.

Importantly, these results are independent of the pair construction method. Both random sampling and BLOSUM-based mutation strategies fail to support stable optimization. Despite their biological plausibility, BLOSUM-guided edits do not yield better results than randomly selected pairs, suggesting that small local mutations are not high quality samples or are insufficiently aligned to provide a clear training signal for DPO.

High-Stability-Only DPO Without Prompts. We also investigate a simplified setting where the model is trained on preference pairs drawn only from the high-stability bucket, with no control tags included in the prompt. This mirrors our first SFT experiment but replaces supervised labels with implicit pairwise preferences. While this setup initially appears stable—generations remain coherent after one epoch—it still fails to show any improvement in stability. Continued training again results in degradation: sequences become repetitive or structurally implausible, and pLDDT scores decline. Once again, both BLOSUM-based and randomly sampled pairs yield similar failure modes.

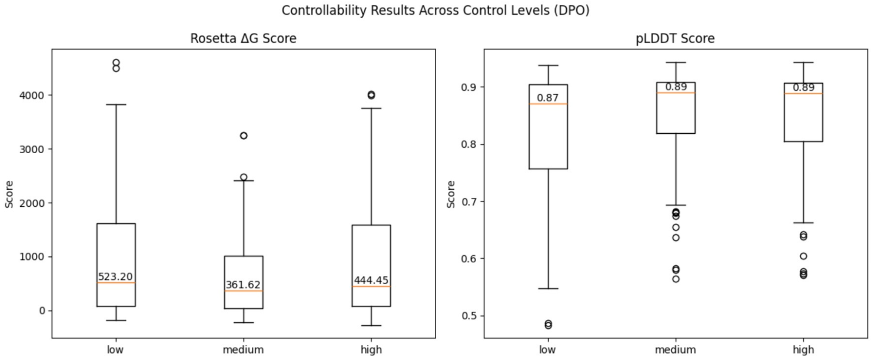

Figure 4: Distributions of predicted Rosetta ΔG for generations under each stability prompt after 1 epoch of DPO with full stability tags. Lower scores indicate higher predicted stability.

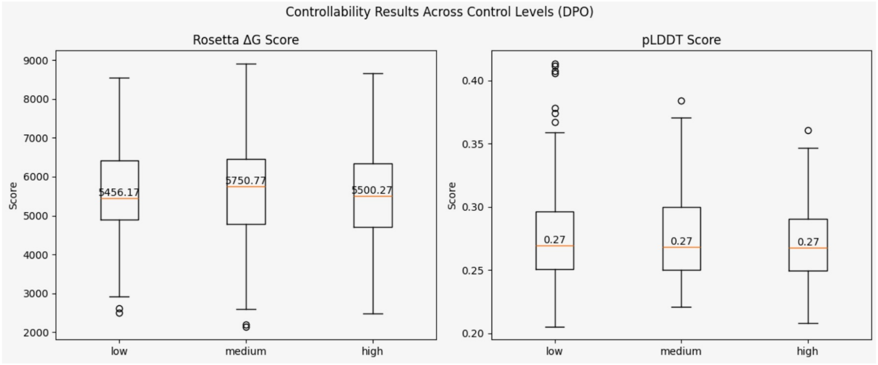

Figure 5: Distributions of predicted Rosetta ΔG for generations under each stability prompt after 2 epochs DPO with full stability tags, exhibiting model collapse. Lower scores indicate higher predicted stability.

Summary. Across all DPO experiments, we find that models exhibit no meaningful learning signal in early epochs and experience sharp degradation with further training. These effects hold across prompt-conditioned and unprompted regimes, and regardless of whether preference data was constructed via random sampling or BLOSUM-guided mutations. These findings highlight the brittleness of preference-based fine-tuning in protein modeling and suggest the need for tighter pair alignment, stronger supervision, and improved optimization strategies.

5 Discussion and Future Work

Using SFT, we were able to improve protein generation stability using both implicit and explicitly labeled fine-tuning. Moreover, we demonstrate that our training methods did not degrade the model's general performance. This result is encouraging, since it indicates that the model has capacity for learning bio-physical traits regarding proteins. Moreover, it is a relatively stable methodology, enabling easy continued training.

Our initial approach ambitiously attempted to apply Direct Preference Optimization (DPO) to learn new control tags from scratch, aiming to steer the model toward generating sequences conditioned on stability tags through preference-based fine-tuning alone. However, this proved challenging: DPO is best viewed as a mechanism for aligning preferences within a distribution the model already understands, rather than instilling entirely new conditioning behavior. We observed clear indicators of model collapse when performing cold-start DPO. With a limited preference dataset (1000 pairs), the model struggled to reliably generate sequences that respected the control tag, confirming that DPO alone was likely insufficient to achieve strong controllability.

We suspect this is because the model has not yet learned meaningful semantic relationships between the stability labels and the subsequent sequence. Thus, the contrastive learning method is learning priorities between meaningless labels. This is the largely accepted reason for why DPO works better on pre-trained features.

We also believe that it is worth noting that the ΔG ranges assigned to each stability label is not uniform in size. The probability distribution over stability scores is roughly Gaussian. During labeling, we assign each third of the distribution mass to each label. This means that the ”high” and ”low” labels have a greater range over of ΔG scores than ”medium.” We suspect that this makes it easier for the model to distinguish between ”high” and ”low” scores than ”medium”, since the ”medium” scores are sequeezed between ”high” and ”low.” We can see potential evidence of this in Figure 4, where the high stability score produces sequences with lower ΔG than low. However, the ΔG distributions for the two labels are not sufficiently distinct to conclusive determine whether our explanation is correct.

In future work, we would like to explore other bio-physical traits to re-use our methodology on. This would act as both a validation of our SFT findings but also an opportunity to successfully employ DPO. We believe that domains where we have a stronger or more reliable scoring oracle could provide more training signal for our fine-tuning algorithms.

We would also like to continue exploring pairing strategies for DPO. Rather than create DPO pairs from scratch, we believe that using experimental data could offer an opportunity to provide the algorithm high quality signal. For instance, evolutionary data could provide clear examples of protein mutations which are both biologically plausible but also known to be beneficial for evolutionary fitness. We also note that the methodology in Widatalla et al. [2024] uses experimentally validated pair data, which proves to be a successful strategy.

Finally, we are also interested in methods for improving DPO stability. DPO offers an exciting avenue towards fine-tuning models for specific tasks, but suffers from high training sensitivity. We are interested in better understanding what DPO does and whether it can be adapted into a more robust formula for effectively fine-tuning protein language models or other biological sequence models.

6 Conclusion

In this work, we introduced a framework for extending protein language models (pLMs) with new control tags, focusing on protein folding stability as a test case. Our hybrid pipeline that combines supervised fine-tuning (SFT) with Direct Preference Optimization (DPO) aims to enable prompt-based control over stability by leveraging both labeled data and dynamically generated preference pairs.

While our current results highlight key challenges, including the limitations of DPO when applied in isolation and the need for stronger SFT baselines, we have outlined a clear path forward with improvements to data quality, fine-tuning strategy, and curriculum learning.

More broadly, we envision this approach as a general template for extending any pLM to support new controllable properties, provided a sufficiently reliable oracle exists to guide learning. Future work will explore richer control spaces, better proxy metrics, and, ultimately, experimental validation to close the loop between in silico optimization and wet-lab results.

Team Contributions

All work for this project was done by Nahum Maru

References

Geraldene Munsamy, Ramiro Illanes-Vicioso, Silvia Funcillo, Ioanna T. Nakou, Sebastian Lindner, Gavin Ayres, Lesley S. Sheehan, Steven Moss, Ulrich Eckhard, Philipp Lorenz, and Noelia Ferruz. Conditional language models enable the efficient design of proficient enzymes. *bioRxiv*, 2024. doi: 10.1101/2024.05.03.592223. URL https://www.biorxiv.org/content/early/2024/05/05/2024.05.03.592223.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36:53728–53741, 2023.

Alexander Rives, Joshua Meier, Tom Sercu, Siddharth Goyal, Zeming Lin, Jason Liu, Demi Guo, Myle Ott, C. Lawrence Zitnick, Jerry Ma, and Rob Fergus. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proceedings of the National Academy of Sciences*, 118(15):e2016239118, 2021. doi: 10.1073/pnas.2016239118. URL https://www.pnas.org/doi/abs/10.1073/pnas.2016239118.

Talal Widatalla, Rafael Rafailov, and Brian Hie. Aligning protein generative models with experimental fitness via direct preference optimization. *bioRxiv*, 2024. doi: 10.1101/2024.05.20.595026. URL https://www.biorxiv.org/content/10.1101/2024.05.20.595026v1.

Jason Yang, Aadyot Bhatnagar, Jeffrey A Ruffolo, and Ali Madani. Conditional enzyme generation using protein language models with adapters. *arXiv preprint arXiv:2410.03634*, 2024.