Sparse Autoencoders in Protein Engineering Campaigns: Steering and Model Diffing

Anonymous Authors[1]

Abstract

Protein Language Models (pLM) have proven versatile tools in protein design, but their internal workings remain difficult to interpret. Here, we implement a mechanistic interpretability framework and apply it in two scenarios. First, by training sparse autoencoders (SAEs) on the model activations, we identify and annotate features relevant to enzyme variant activity through a two-stage process involving candidate selection and causal intervention. During sequence generation, we steer the model by clamping or ablating key SAE features, which increases the predicted enzyme activity. Second, we compare pLM checkpoints before and after three rounds of Reinforcement Learning (RL) by examining sequence regions with high divergence of per-token log-likelihood, detecting the residues that most align with higher predicted affinities.

1. Introduction

End-to-end differentiable models are complex nonlinear functions $f : X \to Y$ that map an input space X to an output space Y. These mapping functions are essentially black boxes, making it difficult to explain *how* and *why* a model ends up making a particular decision. Protein language models (pLMs), are no exception, but despite their hermetic nature, pLMs must have nevertheless learned some complex sequence-to-function relationships, as evidenced by their versatility and state-of-the-art performance in tasks ranging from protein folding (Lin et al., 2023a) to protein design(Yang et al., 2024; Madani et al., 2023; Bhatnagar et al., 2025), including distant yet catalytically efficient enzymes (Munsamy et al., 2022; Madani et al., 2023; Johnson et al., 2023; Parsan et al., 2025).

Mechanistic interpretability aims to provide a detailed analysis of the mechanisms underlying the predictions of deep learning models. Sparse Autoencoders (SAE) in particular have recently emerged as a relevant tool to extract interpretable features and compose them, for the study of internal circuits from LLMs (Marks et al., 2024). In the field of protein research, we are witnessing applications for pLMs with promising outcomes, especially for the understanding of encoder-only pLMs (Parsan et al., 2025; Simon & Zou, 2024; Adams et al., 2025; Garcia & Ansuini, 2025).

SAE models consist of an encoder-decoder architecture that learns to produce intermediate activations of higher dimensionality 1, incentivized to be sparse through the training process. In particular, the encoder transforms an input x into an intermediate vector through a function f, ensuring the activations are sparse (i.e., present few non-zero features) by applying a *BatchTopK* activation that retains the $k \times n$ largest entries of the SAE latent within each batch, zeroing out all the others (Bussmann et al., 2024) (Eq. 1). The decoder learns to reconstruct the activations x as output (Eq. 2), by applying a training loss that is formulated to both reconstruct the model activation by the mean square error of the vector x and x (Eq. 3) with the auxiliary loss that ensures sparsity (Eq. 4):

$$\mathbf{f}(\mathbf{x}) = \mathrm{BatchTopK}\left(\mathbf{W}_{\mathrm{enc}}\mathbf{x} + \mathbf{b}_{\mathrm{enc}}\right) \tag{1}$$

$$\hat{\mathbf{x}} = \mathbf{W}_{\mathrm{dec}}\,\mathbf{f} + \mathbf{b}_{\mathrm{dec}} \tag{2}$$

$$\mathcal{L}(\mathbf{x}) = \mathrm{MSE}\left(\mathbf{x}, \hat{\mathbf{x}}\right) + \mathcal{L}_{\mathrm{aux}} \tag{3}$$

$$\mathcal{L}_{\mathrm{aux}} = \mathrm{MSE}\left(\mathbf{e}, \hat{\mathbf{e}}\right) \tag{4}$$

In this work, we investigate the potential of SAEs in the context of decoder-only pLMs. We explore their application for interventions during inference (steering). Additionally, we study the changes induced in the internal representations of the model comparing the checkpoints of ZymCTRL, a conditional pLM, before and after alignment through direct preference optimization (DPO), to understand the position-dependent patterns learned during RL campaigns.

The contributions of this work are threefold:

- We trained a suite of Sparse Autoencoders on $\sim$1 billion tokens from the BRENDA enzyme database. These SAEs can be applied to diverse downstream tasks, such as explainability or enzyme design.

- We developed a protein engineering workflow by fine-tuning these SAEs on α-amylase DMS data, identifying features that correlate with fitness through sparse logistic regression. We implementing causal interventions (feature clamping and ablation) with the goal of steering the model toward the desired fitness.
- We analyze how protein language models evolve under RL alignment by applying model diffing, revealing both localized amino acid preference shifts and broader changes in sequence exploration strategies between pre- and post-alignment checkpoints.

2. Methods

2.1. Activity prediction Oracle and Dataset

Following (Schmirler et al., 2024), we trained an activity prediction oracle by fine-tuning ESM-1v (Meier et al., 2021), with LoRA adapters. Both models were trained to predict the activity of α-amylase variants using publicly available datasets from the Protein Engineering Tournament GitHub repository (Armer et al., 2023). Specifically, the models were trained to predict SAPI values, which represent the ratio of the specific activity of a variant to that of the reference enzyme. Prior to training, we filtered out entries with no recorded activity or with expression below 0.5. The models were trained for 57 epochs using an 80/20 split for training and validation. A batch size of 4 and a learning rate of 3×10^{-4} were applied during training. Learning curves and Spearman correlations are illustrated in Figure A14.

2.2. SAE architecture and Datasets

We trained a suite of sparse autoencoders on approximately 1 billion tokens from the BRENDA enzyme database (Schomburg et al., 2000), injecting them into the residual stream of ZymCTRL before the attention module. Following best practices, we used the BatchTopK activation function during training, which retains only the top-$k \times b$ activations per batch, where b is the batch size.

After pretraining, we fine-tuned each SAE on our Deep Mutational Scanning dataset with a reduced learning rate to prevent overfitting. During training, the batch size was set to 4096, with a learning rate of 3×10^{-4}, using the Adam optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.99$, and an expansion factor of 12. The residual stream dimension is 1280 yielding $1280 \times 12 = 15360$ latents (decoder rows). Layer 26 was chosen based on preliminary results indicating superior performance compared to other insertion points 13.

2.3. Feature Selection and Causal Interventions

To identify the most predictive latent features, we pooled position-wise activations into sequence-level vectors and trained a Sparse Logistic Regression model using the Sklearn implementation (Pedregosa et al., 2011). The resulting coefficient vector has as many entries as decoder columns, with most coefficients being zero. Features with nonzero coefficients were subsequently used for downstream interventions. Interventions were performed at inference time if the target feature was activated during the forward pass. Specifically, clamping involved setting the activations of features identified as positively correlated with the activity, to their maximum observed values in the training set. In contrast, ablation was carried out by setting to zero the features that were negatively correlated with the activity.

2.4. Fine tuning and DPO-alignment

ZymCTRL was fine-tuned on 10,398 protein sequences, as detailed in the model card available on Hugging Face (AI4PD/ZymCTRL). Fine-tuning was performed over 28 epochs with a learning rate of $8 \times 10^{-}5$. Following fine-tuning, the model was aligned using the Weighted DPO framework, as described in (Stocco et al., 2024). The reward function was defined as the mean of three components: (i) predicted activity, (ii) pLDDT (score, and (iii) TM-score (van Kempen et al., 2023) of the esm-fold (Lin et al., 2023b) predicted protein structure. To mitigate reward hacking and sequence length bias, the final reward was weighted using a Gaussian length penalty centered at 425 residues,the typical length of sequences in the DMS dataset.

2.5. Model Diffing

The pipeline described above maps two global properties of an enzyme variant: its predicted activity and the position-wise pooled SAE activations.

To investigate position-dependent sequence–activity relationships, we compare the next-token probability distributions produced by two checkpoints of our model: the base model and the DPO-aligned model at iteration 3, as it showed the highest reward (Figure A2)

At each sequence position, we compute the Kullback–Leibler (KL) divergence between the two models' next-token distributions using the raw, ungapped sequences. For comparison between the two models, we aggregate the KL divergences by aligning the per-position KL divergence by re-indexing based on a multiple sequence alignment (MSA) (Figure A**??**),allowing to compare sequences of different lengths. In particular, MSAs of all generated variants are performed using MAFFT (Katoh et al., 2002) with default settings. We then re-index the per-sequence KL divergence scores onto the MSA coordinate frame, so that each divergence value corresponds to a consistent alignment position across variants.

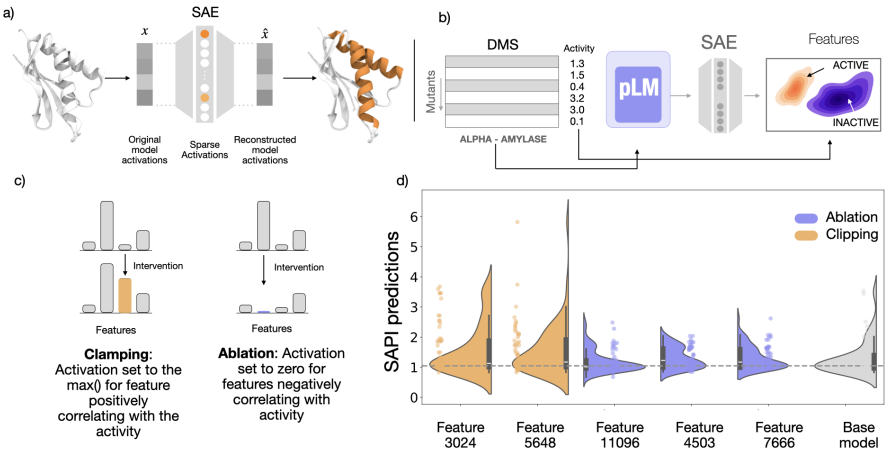

Figure 1. a) Schematic representation of the training process for SAEs. The SAE is inserted between the model's layers. Embeddings x are passed through the encoder-decoder and reconstructed as $\hat{x}$, with sparsity enforced in a higher-dimensional space than the input vector. This may provides a more interpretable representation, as learned features can potentially be correlated with observed features. b) Specific application of SAEs for protein engineering, as exemplified in this work. ZymCTRL (pLM) is fed with DMS data, and correlations between learned features and activity measurements are used to interpret and extract relevant features that are then used to steer the model. c) Steering is performed through clamping and ablation. The resulting effects reveal an increase in the average predicted activity compared to the base model d).

Finally, we select the top MSA columns by average KL divergence. These top-KL positions highlight the residues where the base and DPO-aligned models differ most strongly in their predictive distributions.

3. Experiments and Results

3.1. Steering Interventions for Enzyme Generation

Following Chalnev et al. (Chalnev et al., 2024), we assessed two causal interventions on feature activations during autoregressive generation of enzyme variants. In the *ablation* intervention, whenever a targeted feature naturally activated, its value was set to zero; in the *clamping* intervention, any activation was set to its maximum observed value in the training distribution (Figure 1c). Both methods relied on reconstruction-error terms from a Sparse Autoencoder to preserve sequence quality. As a baseline, we implemented Contrastive Activation Addition (CAA) (Panickssery et al., 2023), which adds a "steering vector" during generation equal to the difference between mean activations of high-activity (> 2.5) versus low-activity (< 2.5) α-amylase classes.

We generated large ensembles under each steering scheme and from the unmodified base model, then predicted their enzymatic activities using our trained oracle. Distributions were compared to the base using the Mann–Whitney U test; only statistically significant shifts were retained for further analysis (Table 1).

Intervention	Median Predicted Activity	p-value vs. Base
Base (no steering)	1.045	—
Ablation	1.051	0.003
Clamping	1.139	< 0.001
CAA	1.058	0.015

Table 1. Median predicted activities and significance of steering interventions compared to the base model.

Clamping produced the largest shift (median +0.13), followed by ablation (+0.07) and CAA (+0.05), confirming that targeted feature manipulations can guide predicted enzyme activity in some cases. More concretely, out of the 45 steering interventions tested (17 ablation, 15 clamping, and 13 CAA), only 11 interventions deviated from the base distribution in a statistically significant way. Of those 11, only 4 interventions (all of them clamping) had a median activity higher than the reference distribution.

3.2. Diffing Dynamics During RL Alignment and Interpreting Model Evolution

We applied DPO for three iterations, consisting of less than 0.1% of the compute used in initial pre-training stage (Ferruz & Höcker, 2022)—to align the model towards higher activity. We generated sequences from both the base and DPO-aligned models, performed MSA to re-index per-token similarity metrics, and computed the KL divergence at each MSA position.

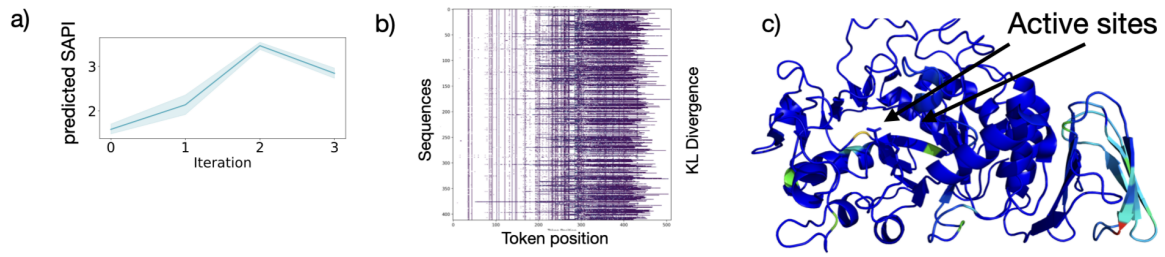

Figure 2. a) Starting from the FT-ZymCTRL, it was aligned with DPO setting the objective of increasing the average predicted SAPI.b) By comparing the two model checkpoints—FT-ZymCTRL and its aligned counterpart DPO-ZymCTRL, and visualizing their output sequences using MSAs, we observe a clear pattern: amino acids with low (near-zero) Kullback-Leibler Divergence (KLD) values (in white) are distinctly separated from regions with higher KLD values. This indicates that the model explores mutations also in regions that are evolutionarily conserved. c) Furthermore, it is possible to visualize KL divergence in 3D using the structure of the reference α-amylase (PDB: 1BAG). The active sites are pointed in the 3d structure,and light blue residues correspond to higher KL divergence.

Inspection of the highest-divergence positions (Fig. 2) revealed two distinct patterns: sparse, discrete substitutions at key residues (vertical columns on key position that span all the enzyme variants), and broader distributional shifts across contiguous regions of the protein .

3.3. Testing and Quantifying AA Transition Patterns

By exploiting the first type of pattern (discrete substitutions at key residues), we can identify positions whose distribution changed the most through the alignment process with the fitness oracle. From this analysis, five positions (94, 99, 130, 277, 285) exhibited the highest divergence. For each site, we constructed two variant sets: one replacing the wild-type residue with the amino acid favored by the base model, and the other using the DPO-aligned model's top prediction. All other residues remained unchanged. We then predicted activities for both sets and computed the mean activity difference for each single-point substitution (Table 2).

Residue Position	AA Transition	Δ Mean Activity
94	I $\rightarrow$ L	0.010
99	D $\rightarrow$ E	**0.716**
130	I $\rightarrow$ L	0.103
277	D $\rightarrow$ Q	0.010
285	S $\rightarrow$ L	**0.946**

Table 2. Activity shifts for single-point mutations informed by base vs. DPO model preferences.

The S$\rightarrow$L substitution at position 285 drove the largest gain (mean +0.946), with D$\rightarrow$E at position 99 yielding +0.716. A moderate improvement was observed for I$\rightarrow$L at 130 (+0.103), whereas transitions at 94 and 277 were effectively neutral (each +0.010). These results demonstrate that a handful of targeted amino acid changes can recapitulate most of the alignment-induced activity enhancements.

4. Discussion and Limitations

Reverse-enegenering to make neural networks human-interpretable is the aim of mechanistinc interpretability (Olah et al., 2018; Meng et al., 2022; Nanda et al., 2023). A key challenge of mechanistic interpretability is identifying the correct units of analysis, that are ideally canonical (irreducibile, indivisible, and complete)(Leask et al., 2025). Due to their properties, SAEs offer intriguing possibilities for interpretability research.

In this work, we explored the application of SAEs in the context of a protein engineering campaign. Specifically, we trained SAEs and extracted features that correlate with an external oracle trained to predict enzyme activity. By ablating and clamping targetted activations, we observed it is possible to deviate the base model distribution, although the effect of a single intervention at a time remains modest. We also computed KL divergences between base and aligned models, to investigate how RL alters the model's internal representations. Through this process, we were able to capture fine-grained differences and identify how individual mutations contributed to measurable improvements in generated sequences.

In future work, we envision (1) combining steering multiple interventions (clamping and ablating) for the engineering of α-amylase variants, and (2) testing base and steered designs experimentally.

Acknowledgements

Will be reported in the accepted version.

Impact Statement

Will be reported in the accepted version.

References

Adams, E., Bai, L., Lee, M., Yu, Y., and AlQuraishi, M. From mechanistic interpretability to mechanistic biology: Training, evaluating, and interpreting sparse autoencoders on protein language models. *bioRxiv*, pp. 2025–02, 2025.

Armer, C., Kane, H., Cortade, D., Estell, D., Yusuf, A., Sanka, R., Redestig, H., Brunette, T., Kelly, P., and DeBenedictis, E. The Protein Engineering Tournament: An Open Science Benchmark for Protein Modeling and Design, 2023. URL https://arxiv.org/abs/2309.09955. Version Number: 2.

Bhatnagar, A., Jain, S., Beazer, J., Curran, S. C., Hoffnagle, A. M., Ching, K., Martyn, M., Nayfach, S., Ruffolo, J. A., and Madani, A. Scaling unlocks broader generation and deeper functional understanding of proteins, April 2025. URL http://biorxiv.org/lookup/doi/10.1101/2025.04.15.649055.

Bussmann, B., Leask, P., and Nanda, N. Batchtopk sparse autoencoders. *arXiv preprint arXiv:2412.06410*, 2024.

Chalnev, S., Siu, M., and Conmy, A. Improving steering vectors by targeting sparse autoencoder features. *arXiv preprint arXiv:2411.02193*, 2024.

Ferruz, N. and Höcker, B. Controllable protein design with language models. *Nature Machine Intelligence*, 4(6): 521–532, 2022.

Garcia, E. N. V. and Ansuini, A. Interpreting and Steering Protein Language Models through Sparse Autoencoders, 2025. URL https://arxiv.org/abs/2502.09135. Version Number: 1.

Johnson, S. R., Fu, X., Viknander, S., Goldin, C., Monaco, S., Zelezniak, A., and Yang, K. K. Computational Scoring and Experimental Evaluation of Enzymes Generated by Neural Networks, March 2023. URL http://biorxiv.org/lookup/doi/10.1101/2023.03.04.531015.

Katoh, K., Misawa, K., Kuma, K.-i., and Miyata, T. Mafft: a novel method for rapid multiple sequence alignment based on fast fourier transform. *Nucleic acids research*, 30(14):3059–3066, 2002.

Leask, P., Bussmann, B., Pearce, M., Bloom, J., Tigges, C., Moubayed, N. A., Sharkey, L., and Nanda, N. Sparse autoencoders do not find canonical units of analysis, 2025. URL https://arxiv.org/abs/2502.04878.

Lin, Z., Akin, H., Rao, R., Hie, B., Zhu, Z., Lu, W., Smetanin, N., Verkuil, R., Kabeli, O., Shmueli, Y., Dos Santos Costa, A., Fazel-Zarandi, M., Sercu, T., Candido, S., and Rives, A. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, March 2023a. ISSN 0036-8075, 1095-9203. doi: 10.1126/science.ade2574. URL https://www.science.org/doi/10.1126/science.ade2574.

Lin, Z., Akin, H., Rao, R., Hie, B., Zhu, Z., Lu, W., Smetanin, N., Verkuil, R., Kabeli, O., Shmueli, Y., et al. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637): 1123–1130, 2023b.

Madani, A., Krause, B., Greene, E. R., Subramanian, S., Mohr, B. P., Holton, J. M., Olmos, J. L., Xiong, C., Sun, Z. Z., Socher, R., Fraser, J. S., and Naik, N. Large language models generate functional protein sequences across diverse families. *Nature Biotechnology*, 41(8):1099–1106, August 2023. ISSN 1087-0156, 1546-1696. doi: 10.1038/s41587-022-01618-2. URL https://www.nature.com/articles/s41587-022-01618-2.

Marks, S., Rager, C., Michaud, E. J., Belinkov, Y., Bau, D., and Mueller, A. Sparse feature circuits: Discovering and editing interpretable causal graphs in language models. *arXiv preprint arXiv:2403.19647*, 2024.

Meier, J., Rao, R., Verkuil, R., Liu, J., Sercu, T., and Rives, A. Language models enable zero-shot prediction of the effects of mutations on protein function. *bioRxiv*, July 2021. doi: 10.1101/2021.07.09.450648. URL http://dx.doi.org/10.1101/2021.07.09.450648.

Meng, K., Bau, D., Andonian, A., and Belinkov, Y. Locating and editing factual associations in gpt. *Advances in neural information processing systems*, 35:17359–17372, 2022.

Munsamy, G., Lindner, S., Lorenz, P., and Ferruz, N. Zymctrl: a conditional language model for the controllable generation of artificial enzymes. In *NeurIPS machine learning in structural biology workshop*, 2022.

Nanda, N., Rajamanoharan, S., Kramar, J., and Shah, R. Fact finding: Attempting to reverse-engineer factual recall on the neuron level. In *Alignment Forum*, pp. 6, 2023.

Olah, C., Satyanarayan, A., Johnson, I., Carter, S., Schubert, L., Ye, K., and Mordvintsev, A. The building blocks of interpretability. *Distill*, 3(3):e10, 2018.

Panickssery, N., Gabrieli, N., Schulz, J., Tong, M., Hubinger, E., and Turner, A. M. Steering llama 2 via contrastive activation addition. *arXiv preprint arXiv:2312.06681*, 2023.

Parsan, N., Yang, D. J., and Yang, J. J. Towards interpretable protein structure prediction with sparse autoencoders. *arXiv preprint arXiv:2503.08764*, 2025.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Schmirler, R., Heinzinger, M., and Rost, B. Fine-tuning protein language models boosts predictions across diverse tasks. *Nature Communications*, 15(1):7407, 2024.

Schomburg, I., Hofmann, O., Baensch, C., Chang, A., and Schomburg, D. Enzyme data and metabolic information: Brenda, a resource for research in biology, biochemistry, and medicine. *Gene Function & Disease*, 1(3-4):109–118, 2000.

Simon, E. and Zou, J. Interplm: Discovering interpretable features in protein language models via sparse autoencoders. *bioRxiv*, pp. 2024–11, 2024.

Stocco, F., Artigues-Lleixa, M., Hunklinger, A., Widatalla, T., Guell, M., and Ferruz, N. Guiding generative protein language models with reinforcement learning. *arXiv preprint arXiv:2412.12979*, 2024.

van Kempen, M., Kim, S. S., Tumescheit, C., Mirdita, M., Lee, J., Gilchrist, C. L. M., Söding, J., and Steinegger, M. Fast and accurate protein structure search with foldseek. *Nature Biotechnology*, 42(2): 243–246, May 2023. ISSN 1546-1696. doi: 10.1038/s41587-023-01773-0. URL http://dx.doi.org/10.1038/s41587-023-01773-0.

Yang, J., Bhatnagar, A., Ruffolo, J. A., and Madani, A. Conditional Enzyme Generation Using Protein Language Models with Adapters, 2024. URL https://arxiv.org/abs/2410.03634. Version Number: 1.

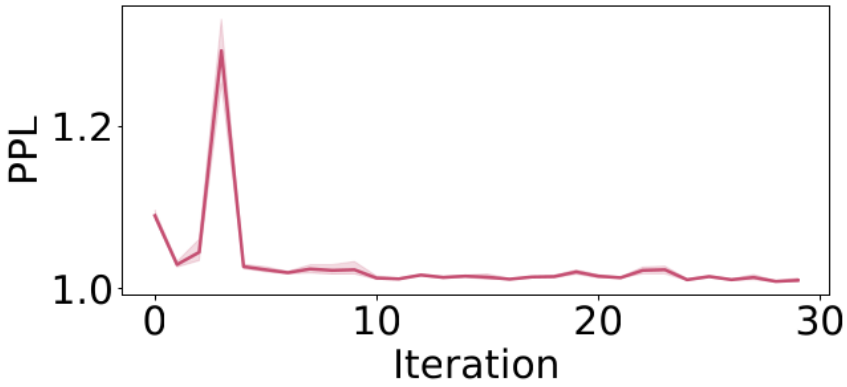

Figure 3. Average sequence perplexity across sequential DPO rounds.

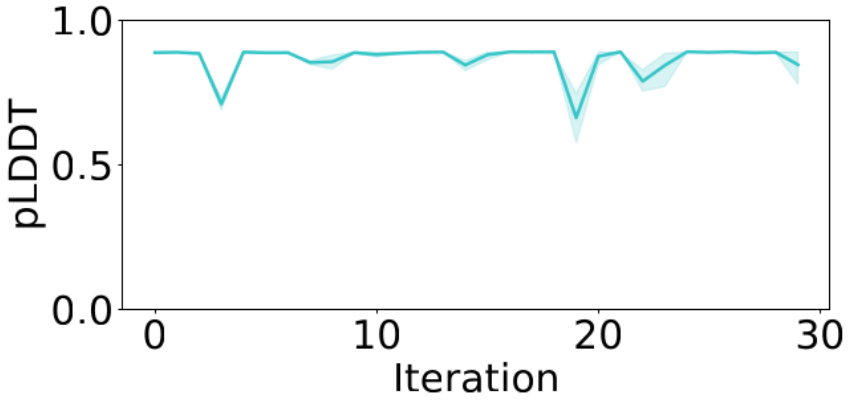

Figure 4. Average pLDDT, as measured by ESMFold, across sequential DPO rounds.

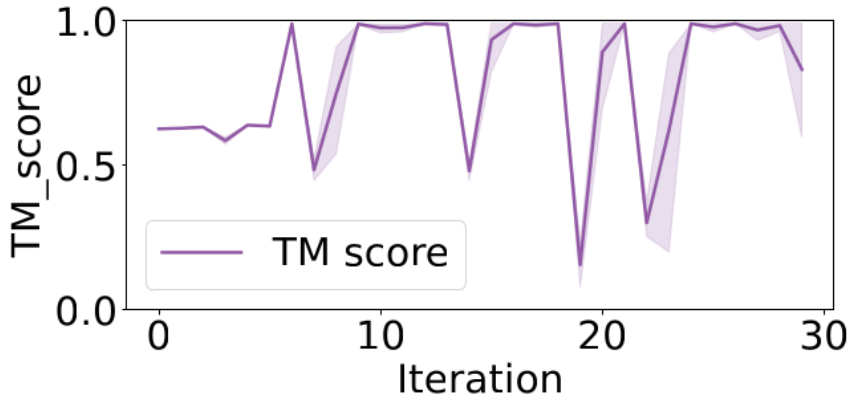

Figure 5. Average TM-score between the reference enzyme and ESMFold-predicted structures during sequential DPO rounds.

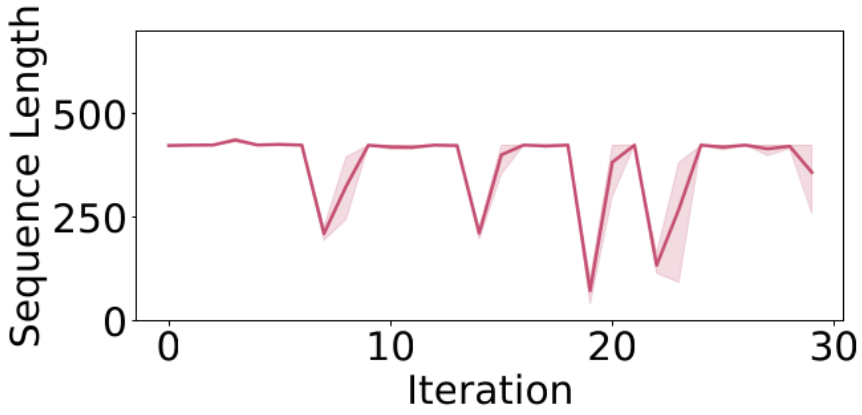

Figure 6. Average sequence length across sequential DPO rounds.

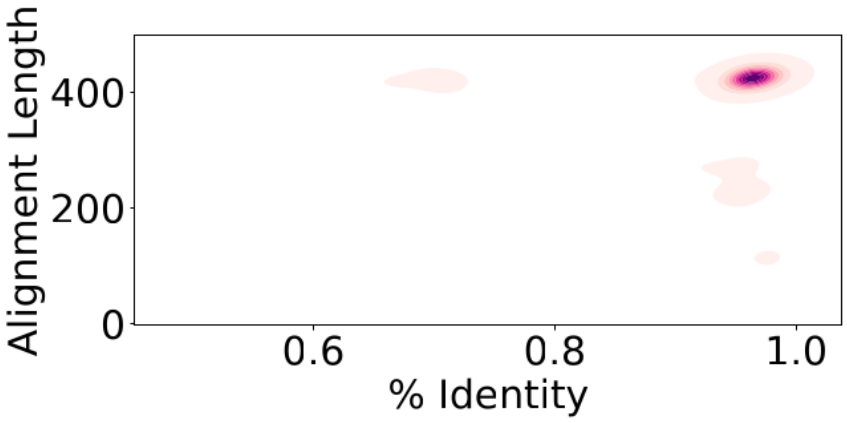

Figure 7. Global distribution of sequence length versus alignment length.

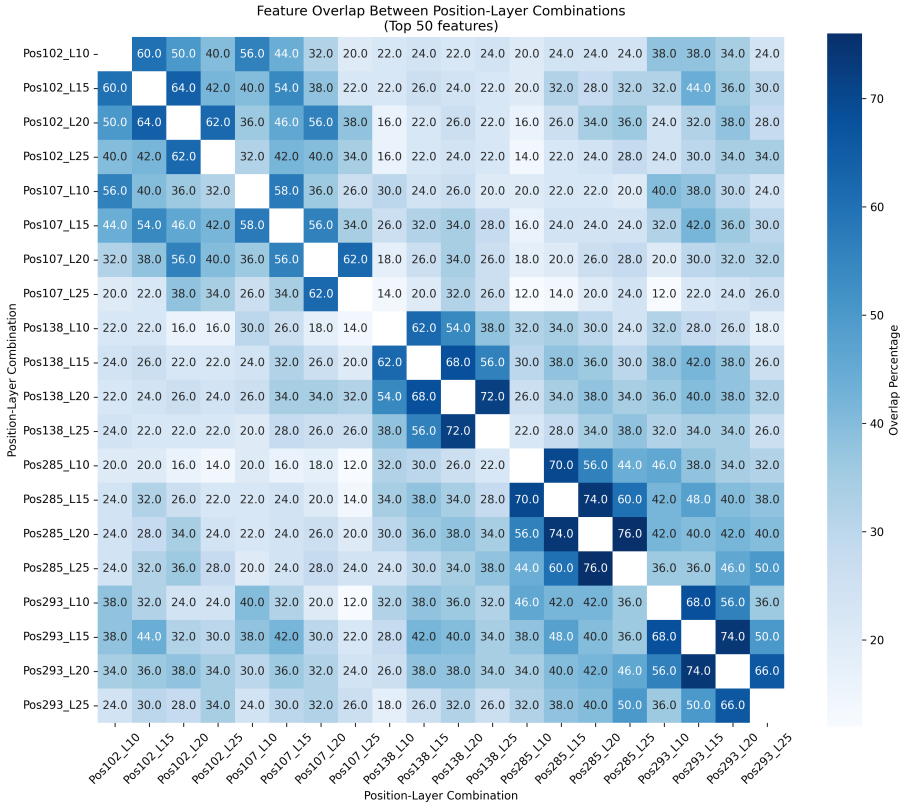

Figure 8. Percentage of overlap between the top ranked features, obtained trough gradient attribution on key-residues

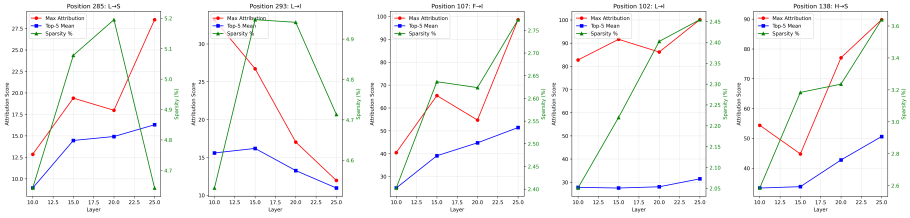

Figure 9. Mean, Maximum and Standard deviation attribution to each layer, for the attribution in the 5 key-residues.

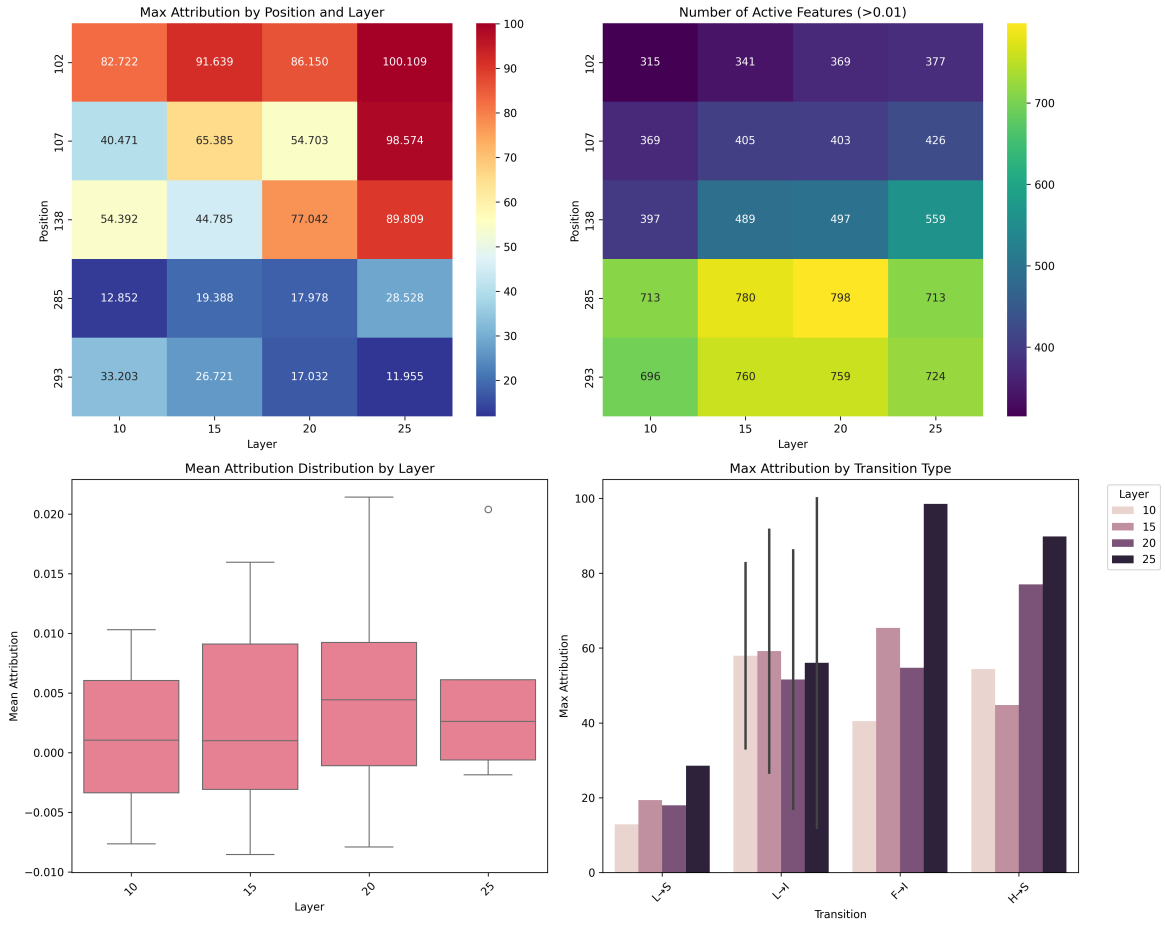

Figure 10. Layer-wise attribution metrics

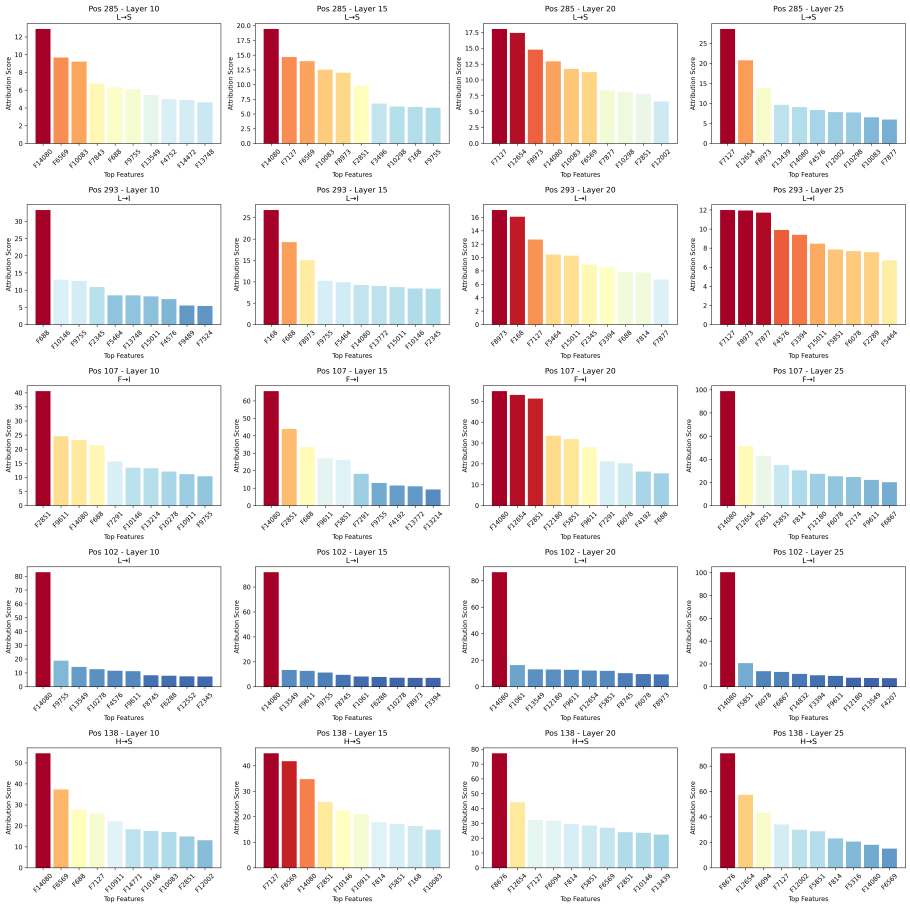

Figure 11. Top features for each position and layer, as measured by gradient attribution on the difference in AA transition logprobs

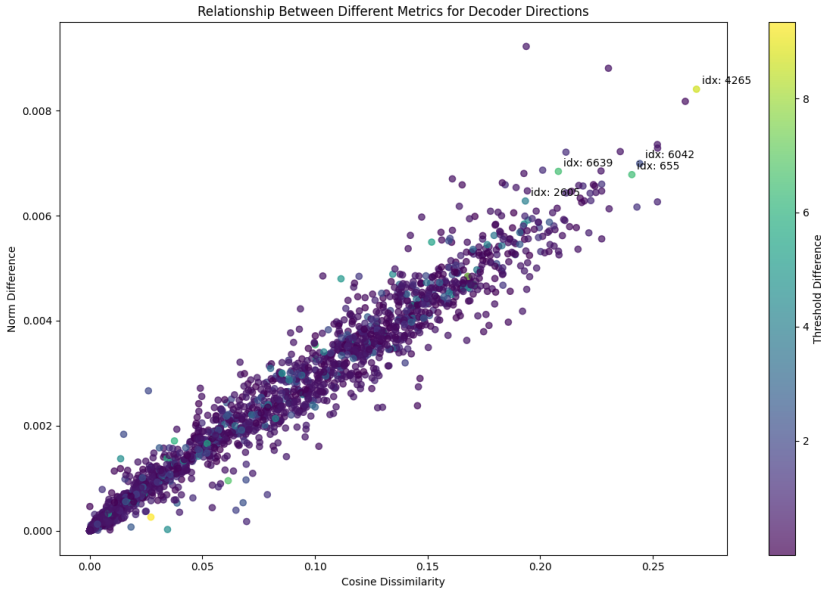

Figure 12. Scatterplot of the cosine similarity and difference in norms of the latents of a SAE before and after finetuning, the points are colored based on the difference in thresholds

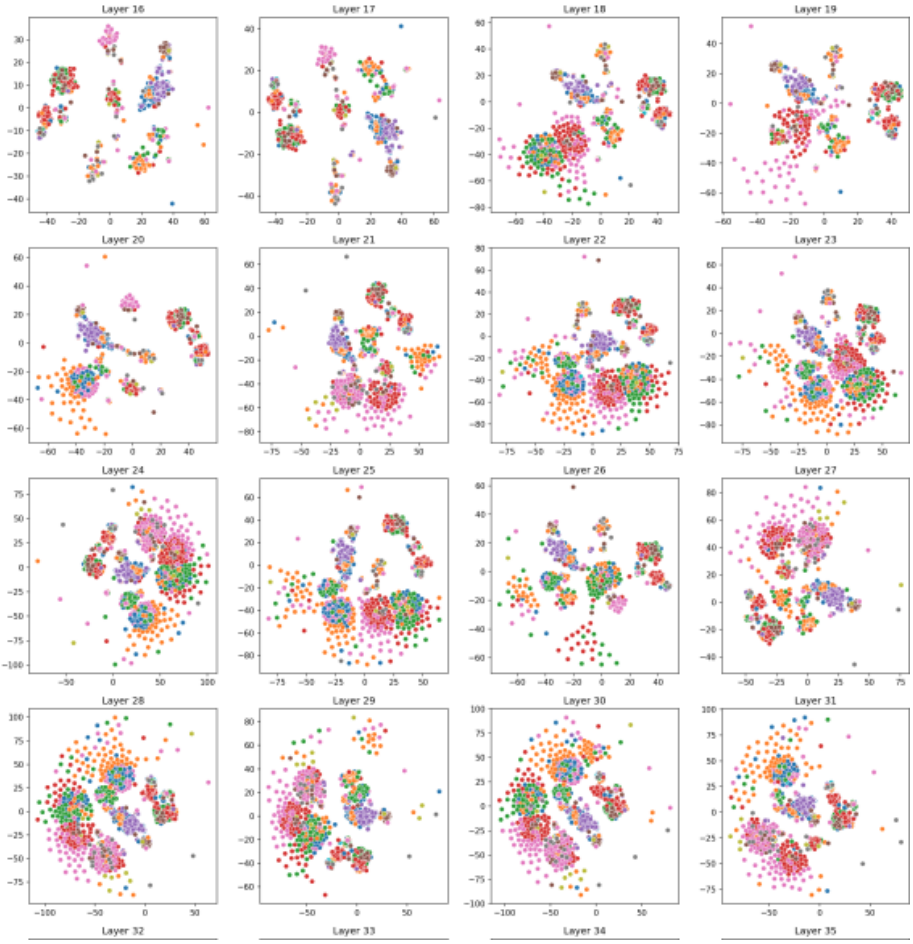

Figure 13. TSNE visualization of the embeddings of DPO sequences at different layers

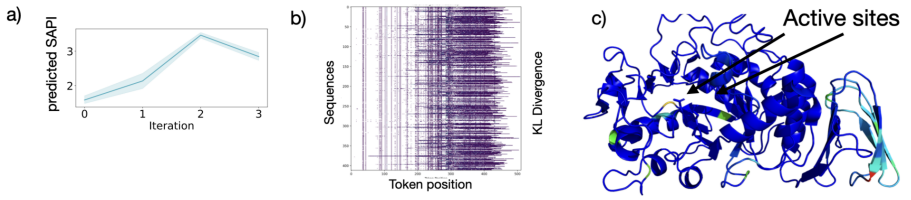

Figure 14. Training curve of esm-1v with Lora Adapter, as reported in Chalnev et al. (Chalnev et al., 2024)