Where to Edit? : Complementary Protein Property Control from Weight and Activation Spaces

Nathan Choi*
California State University, Fullerton
mchoi2159@gmail.com

Armaity Katki*
University of Georgia
armaitykatki@gmail.com

Son Sophak Otra
Northeastern University
sonsopheakotra12@gmail.com

George Flint†
University of California, Berkeley
georgeflint@berkeley.edu

Kevin Zhu†
Algoverse AI Research
kevin@algoverseacademy.com

Sunishchal Dev†
Algoverse AI Research
dev@algoverseairesearch.org

Abstract

Protein language models (PLMs) are powerful tools for protein engineering, but remain difficult to steer toward specific biochemical properties, where small sequence changes can affect stability or function. We adapt two prominent unsupervised editing methods: task arithmetic (TA; specifically, Forgetting via Negation) in weight space and feature editing with a sparse autoencoder (SAE) in activation space. We evaluated their effects on six biochemical properties: net charge at pH 7, hydrophobicity, aromaticity, instability index, molecular weight, and isoelectric point of generations from three PLMs (ESM3, ProGen2-Large and ProLLaMA). Across models, we observe complementary efficacies: TA more effectively controls some properties, while SAE more effectively controls others. Property response patterns show some consistency across models. We suggest that the response pattern of biochemical properties should be considered when steering PLMs.[3]

1 Introduction

Protein language models (PLMs) have emerged as important means for protein engineering, learning patterns directly from amino acid data [Rives et al., 2021, Lin et al., 2023]. While these models have assisted significant advances in protein structure prediction and functional understanding, approaches to steering them remain underexplored in the literature.

In natural language processing (NLP), recent progress in model editing and interpretability has produced methods to modify model behavior without retraining, including task arithmetic (TA) [Ilharco et al., 2022] and sparse autoencoder-based methods (SAEs) [Parsan et al., 2025]. The efficacies of these approaches suggest that learned representations can be decomposed and manipulated to control specific output properties.

Steering protein language models (PLMs) presents distinct challenges compared to text generation. In biology, generated sequences must remain functional - changes must preserve valid protein folding

*Equal contribution. Primary authors.

†Senior authors.

[3]**Code**: https://github.com/Ulton321/Protein-Language-Model-Steering

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

and maintain essential biological and chemical properties. Unlike text, where creative variation is acceptable, small changes in amino acid sequences can heavily disrupt structure or function [Rives et al., 2021]. Additionally, important biological properties like charge or hydrophobicity are often intertwined within the model's internal representations, making it difficult to isolate and modify one property without unintentionally altering others [Ilharco et al., 2022, Simon and Zou, 2024].

This work investigates whether PLMs can be steered toward target properties using unsupervised editing methods. We study two steering techniques, TA and SAEs, on three PLM models: ESM3, ProGen2-Large, and ProLLaMA. We focus on six biologically meaningful properties: net charge at pH 7, hydrophobicity using GRAVY (Grand Average of Hydropathy–the average Kyte–Doolittle hydropathy per residue, measuring a sequence's overall hydrophobic vs. hydrophilic character), aromaticity, instability index, molecular weight, and isoelectric point.

Our study is guided by the following questions: (1) Can TA or SAEs steer PLMs toward sequences with desired properties? (2) How do these methods compare in efficacy and stability?

To answer these questions, we fine-tune the models on non-examples of a target property and invert the differences in weights–a task arithmetic process called "forgetting via negation" introduced in Ilharco et al. [2022]. We also train SAEs to identify latent directions aligned with the biochemical properties and perform linear edits in activation space. We then identify which steering method is more effective for each property.

By bridging controllability techniques from NLP to protein design, this work contributes to a general-purpose framework for interpretable, property-guided biological sequence generation, while accounting for the dual-use risk surface of PLMs and broader misuse vectors [Ekins et al., 2023, Pannu et al., 2025].

2 Related work

Protein language models (PLMs) have allowed for large-scale unsupervised learning of biological sequence representations. Rives et al. [2021] trained transformer models on protein sequences and showed that they encode structural and functional information. Lin et al. [2023] further demonstrated that PLMs trained on large-scale evolutionary data can accurately predict detailed protein structures. These works are the foundation for the use of PLMs such as ESM in generative protein engineering.

Ilharco et al. [2022] introduced TA as a mechanism to modify the behavior of the model through vector operations in weight space. A task vector is computed by subtracting the weights of a base model from a fine-tuned model, and applying this vector to a new model transfers or removes specific capabilities. We apply this method to biological domains by fine-tuning on nonexamples of a property and using vector negation to steer model outputs toward the desired property.

Parsan et al. [2025] used SAEs to reveal interpretable directions in neural sequence model activations. Simon and Zou [2024] introduced INTERPLM, applying SAEs to PLMs and showing alignment with biochemical properties. Banerjee et al. [2025] advanced this by automating neuron labeling and modulation to guide protein generation. Ackerman [2024] bridged activation- and weight-based control by tuning activation vectors into model weights. Complementing these, Lv et al. [2024] proposed ProLLaMA, a protein language model enabling multi-property control, serving as a strong baseline. These works inform our SAE-based activation steering to shift protein outputs toward desired properties. Unlike prior SAE applications on PLMs, we cast SAE feature editing as an unsupervised steering primitive and systematically compare it to weight-space task arithmetic across three PLMs and six properties [Simon and Zou, 2024, Banerjee et al., 2025].

Although techniques for modifying large models have advanced in NLP, their application to biological sequence generation remains underexplored. In this work, we examine how such methods translate to protein models, where generated sequences must preserve biological validity.

3 Methods

We evaluate three pretrained protein language models: **ESM3** [Rives et al., 2021], **ProGen2-large** [Nijkamp et al., 2022], and **ProLLaMA**.[4] For all models, we generate 500 protein sequences of length

[4]https://huggingface.co/GreatCaptainNemo/ProLLaMA

100 as the standard evaluation setting. Baseline property distributions are established by scoring these generated sequences with Biopython's `ProteinAnalysis`. For fine-tuning experiments, we follow the PLMInterp protocol and use the `train` split from the Hugging Face `protolyze/plminterp` dataset [Parsan et al., 2025], which we use without modification.

3.1 Fine-Tuning

Fine-tuning serves as a supervised steering baseline. For each biochemical property, we curate subsets of the PLMInterp dataset containing 1,000–2,000 sequences, with average sequence lengths between 100–150 amino acids. Each base model is fine-tuned separately on sequences enriched for the target property. After training, we sample the new sequences and compute property scores.

3.2 Task Arithmetic (Forgetting via Negation)

Task Arithmetic (TA) is implemented as weight-space manipulation. Let θ_{base} denote the weights of a base model, and $\theta_{\text{finetuned}}$ those of the same model fine-tuned on non-examples. The task vector is defined as $\Delta\theta = \theta_{\text{finetuned}} - \theta_{\text{base}}$. We apply *Forgetting via Negation* by subtracting this vector from the base model: $\theta_{\text{steered}} = \theta_{\text{base}} - \Delta\theta$. This operation steers the model away from non-example behavior and toward the desired biochemical property without requiring additional retraining.

3.3 Sparse Autoencoders

As a third axis of model steering, we implement latent-space manipulation via Sparse Autoencoders (SAEs). Unlike fine-tuning or weight-space arithmetic, SAE-based steering operates directly on hidden activations, offering a lightweight and interpretable alternative. We train SAEs on the intermediate representations of ESM3, progen, and prollama and then using activations from the final transformer layer across 10,000 sequences sampled from the PLMInterp training split.

The SAE architecture compresses these activations into sparse latent codes using L1 regularization, encouraging disentanglement of biochemical properties. Each latent unit is evaluated for correlation with target properties (e.g., hydrophobicity, aromaticity, charge at pH 7, instability index, molecular weight, and isoelectric point), as computed by Biopython's `ProteinAnalysis`. At inference time, we steer generation by modifying specific latent dimensions and decoding the altered representations back through the SAE decoder into model activations, which are then used by the original model to produce sequence logits.

This approach enables property-specific control without retraining the base model. Compared to fine-tuning and Task Arithmetic, SAE steering offers finer granularity and interpretability, allowing us to isolate and edit biologically meaningful features. We evaluate the impact of latent edits by generating new sequences and analyzing shifts in property distributions. 3.

4 Results

We report results from 500 generated sequences (each 100 amino acids long) with 95% confidence intervals for each property and model. Below we summarize effects per model, then synthesize cross-model patterns. All statements refer directly to Tables 1–3.

ESM3. SAE provides the strongest positive shifts for charge and isoelectric point, while TA yields the largest gain in hydrophobicity. Neither TA nor SAE achieves the intended decrease in aromaticity, and both increase the instability index relative to Base (less stable). Molecular weight shows conflicting scales across methods (TA substantially below Base, SAE far above), suggesting a units or preprocessing discrepancy; we therefore treat ESM3 molecular-weight outcomes as inconclusive.

ProGen2-Large. TA most strongly increases charge and pI, whereas SAE is best for hydrophobicity; fine-tuning is generally weaker on these targets. TA and SAE also raise molecular weight above Base. Because the Base entries for aromaticity and instability appear malformed in the table, we refrain from interpreting direction on those two properties for ProGen2.

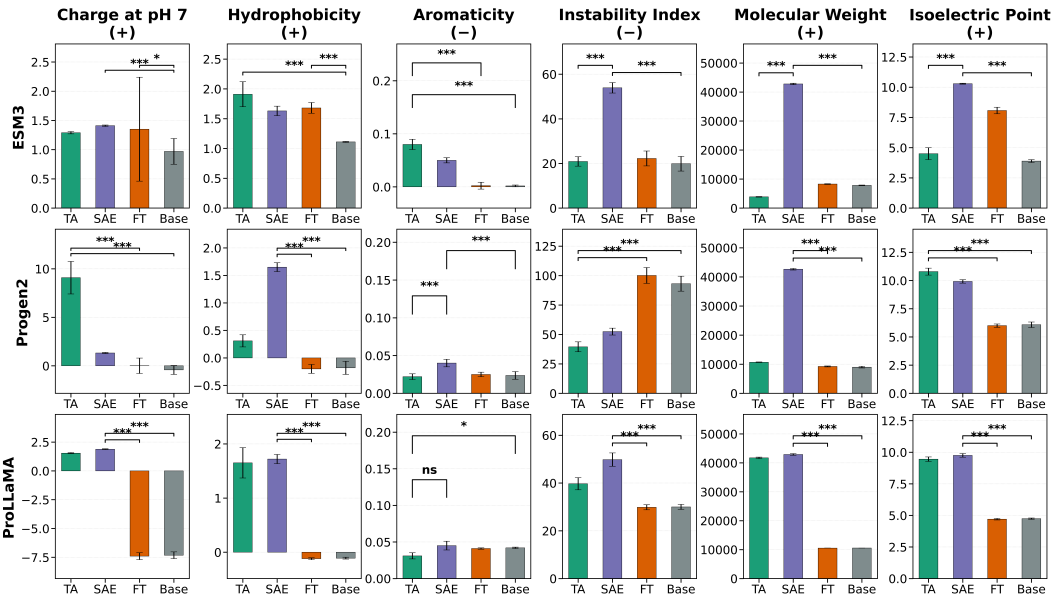

Figure 1: Effects of Task Arithmetic (TA), Sparse Autoencoder (SAE), Fine-Tuned, and Base across six biochemical properties for ESM3, ProGen2, and ProLLaMA. Bars show means over 500 sequences of length 100; error bars = 95% CI. Steering direction is noted (+) and (-). Brackets show pairwise significance (ns, *, **, ***).

ProLLaMA. SAE achieves the largest improvements in charge, hydrophobicity, and pI (with TA close on hydrophobicity). Aromaticity shows a small method-dependent trade-off (TA slightly reduces it; SAE slightly increases it). As with ESM3, both TA and SAE increase the instability index. TA and SAE markedly increase molecular weight relative to Base and fine-tuning.

Trends across models. Weight- and activation-space editing reliably achieve the desired increases in charge and pI, with TA strongest in ProGen2 and SAE strongest in ESM3/ProLLaMA. Hydrophobicity generally favors SAE, except in ESM3 where TA leads. Aromaticity is not uniformly suppressible (ESM3 moves in the wrong direction; ProLLaMA shows small, opposing effects by method). Instability tends to increase under both TA and SAE. Molecular weight typically rises under TA/SAE (ProGen2/ProLLaMA), but ESM3 results are inconsistent.

5 Discussion

Our results indicate that biochemical properties respond differently to steering techniques. In line with our objectives, charge, pI, and (often) hydrophobicity move in the desired directions under editing, while aromaticity and the instability index are harder to control. We attribute the "easy wins" to properties that are largely compositional and therefore linearly available in the model's representation once amino-acid statistics are well learned [Rives et al., 2021, Lin et al., 2023]. In contrast, aromaticity (multifunctional and low-frequency) and instability (a composite, position/dipeptide-dependent score) are not well captured by a single latent axis, making them less responsive to broad edits [Guruprasad et al., 1990, Kyte and Doolittle, 1982]. For TA, this is heightened in low-data clinical contexts, where non-examples are easier to obtain than positive exemplars.

Task Arithmetic (TA) perturbs the model in weight space along a single task vector and is most effective when the target signal is broadly distributed and approximately linear (e.g., enriching Lys/Arg and reducing Asp/Glu for ↑charge and ↑pI) [Ilharco et al., 2022]. This matches our strongest TA gains on charge and pI, particularly in ProGen2. Sparse autoencoders (SAEs) instead factorize activations into sparse, more monosemantic directions; editing those directions exposes controllable knobs for entangled residue-pattern biases such as hydrophobicity [Parsan et al., 2025, Simon and Zou, 2024, Villegas Garcia and Ansuini, 2025]. Consistent with this, SAE often outperforms TA

on GRAVY in ProGen2 and ProLLaMA, while TA remains competitive or best on ESM3. Recent work that translates activation-space edits into weight-space adjustments further suggests a principled bridge between these regimes [Ackerman, 2024].

Layer profiling of PLMs shows that local chemistry emerges early and accumulates through middle layers, while more global, structure/function signals consolidate from middle to later layers [Rives et al., 2021, Lin et al., 2023]. TA's layer-global perturbation is therefore well matched to properties represented across many layers (charge, pI). SAEs, by contrast, let us target the specific blocks where a concept is crisply encoded; prior PLM SAE work reports thousands of interpretable features spanning middle/late blocks that align with biochemical attributes, which accords with our SAE gains on GRAVY and several pI/charge cases [Parsan et al., 2025, Simon and Zou, 2024, Villegas Garcia and Ansuini, 2025].

Prior work suggests that fine-tuning performs best when the target property is locally encoded, chemically distinct, and provides a stable optimization signal (e.g., charge, aromaticity, instability) [Rives et al., 2021, Guruprasad et al., 1990]. It tends to struggle on properties that are entangled, globally distributed, or in tension with other constraints (e.g., hydrophobicity, molecular weight, pI) [Kyte and Doolittle, 1982, Expasy Bioinformatics Resource Portal, 2022, Villegas Garcia and Ansuini, 2025]. In our experiments, we observe partial alignment with these expectations: fine-tuning is sometimes helpful for charge (ESM3) but comparatively weaker or even counter-directional on hydrophobicity (ProGen2) and charge (ProLLaMA). These mixed outcomes reinforce the value of unsupervised editing as a complementary tool rather than a replacement.

We find the following guidline: favor TA when the target is largely compositional or linearly separable in existing representations (charge, pI), and favor SAE when the target exists but is entangled across features (GRAVY). For composite or rare objectives (instability, aromaticity), layer- and feature-specific edits are needed; combining SAE-identified units with small weight-space adjustments may reduce unwanted co-movements [Ackerman, 2024, Parsan et al., 2025].

Limitations. Our analysis is sequence-level and property-driven; we did not localize causality layer-by-layer. Future work should pair per-layer SAE edits with activation patching and controlled ablations to identify the mediating blocks for each property [Parsan et al., 2025, Simon and Zou, 2024], and incorporate structure-aware readouts alongside sequence-level scores [Rives et al., 2021, Lin et al., 2023].

Broader Impacts. This work develops techniques for steering protein language models toward specific biological properties, enhancing controllability in protein design and accelerating bioengineering. Positive impacts include enabling targeted sequence generation for drug discovery and enzyme optimization, and constraining outputs to safer property ranges (e.g., avoiding extreme charge or hydrophobicity) to reduce risks like aggregation or toxicity. Misuse, such as the generation of harmful peptides, remains a concern.

6 Conclusion

We show that weight- and activation-based editing models yield complementary efficacies in steering six biochemical properties with the Evolutionary Scale Modeling protein language model, and suggest that different biochemical properties are encoded in distinct representational spaces.

References

Christopher M. Ackerman. Representation tuning. *arXiv preprint arXiv:2409.06927*, 2024. URL `https://arxiv.org/abs/2409.06927`.

Arjun Banerjee, David Martinez, Camille Dang, and Ethan Tam. Automated neuron labelling enables generative steering and interpretability in protein language models. *arXiv preprint arXiv:2507.06458*, 2025. URL `https://arxiv.org/abs/2507.06458`.

Sean Ekins, Maximilian Brackmann, Cédric Invernizzi, and Filippa Lentzos. Generative artificial intelligence–assisted protein design must consider repurposing potential. *GEN Biotechnology*, 2(4):296–300, 2023. doi: 10.1089/genbio.2023.0025. URL `https://www.liebertpub.com/doi/10.1089/genbio.2023.0025`.

Expasy Bioinformatics Resource Portal. Protparam documentation: Molecular weight computation and related properties. https://web.expasy.org/protparam/protparam-doc.html, 2022. Accessed: 2025-08-22.

K Guruprasad, B V Reddy, and M W Pandit. Correlation between stability of a protein and its dipeptide composition: a novel approach for predicting in vivo stability of a protein from its primary sequence. *Protein Engineering*, 4(2):155–161, 1990. doi: 10.1093/protein/4.2.155.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. In *Proceedings of the 39th International Conference on Machine Learning*, pages 8576–8598, 2022.

Jack Kyte and Russell F Doolittle. A simple method for displaying the hydropathic character of a protein. *Journal of Molecular Biology*, 157(1):105–132, 1982. doi: 10.1016/0022-2836(82)90515-0.

Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, Allan dos Santos Costa, Maryam Fazel-Zarandi, Tom Sercu, Salvatore Candido, and Alexander Rives. Evolutionary-scale prediction of atomic level protein structure with a language model. *bioRxiv*, 2023. URL https://www.biorxiv.org/content/10.1101/2022.07.20.500902v2.

Liuzhenghao Lv, Zongying Lin, Hao Li, Yuyang Liu, Jiaxi Cui, Calvin Yu-Chian Chen, Li Yuan, and Yonghong Tian. Prollama: A protein large language model for multi-task protein language processing. *arXiv preprint arXiv:2402.16445*, 2024. URL https://arxiv.org/abs/2402.16445.

Erik Nijkamp, Jeffrey A. Ruffolo, Eli N. Weinstein, Nikhil Naik, and Ali Madani. Progen2: Exploring the boundaries of protein language models. *arXiv preprint arXiv:2206.13517*, 2022. URL https://arxiv.org/abs/2206.13517.

Jassi Pannu, Dan Bloomfield, Ross MacKnight, Moritz S. Hanke, Alex Zhu, et al. Dual-use capabilities of concern of biological ai models. *PLOS Computational Biology*, 21(5):e1012975, 2025. doi: 10.1371/journal.pcbi.1012975. URL https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1012975.

Nithin Parsan, David J. Yang, and John J. Yang. Towards interpretable protein structure prediction with sparse autoencoders. In *International Conference on Learning Representations*, 2025.

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. URL https://www.pnas.org/doi/10.1073/pnas.2016239118.

Elana Simon and James Zou. Interplm: Discovering interpretable features in protein language models via sparse autoencoders. *arXiv preprint arXiv:2412.12101*, 2024. URL https://arxiv.org/abs/2412.12101.

Edith N. Villegas Garcia and Alessio Ansuini. Interpreting and steering protein language models through sparse autoencoders. *bioRxiv*, 2025. URL https://arxiv.org/abs/2502.09135.

A Result Tables

Property	TA	SAE	Fine-Tuned	Base
charge at pH 7	mean=1.29 std=1.60 95% CI=±0.020	mean=1.41 std=1.98 95% CI=±0.012	mean=1.35 std=10.20 95% CI=±0.89	mean=0.97 std=8.28 95% CI=±0.22
hydrophobicity	mean=1.91 std=1.10 95% CI=±0.21	mean=1.63 std=1.22 95% CI=±0.08	mean=1.68 std=0.99 95% CI=±0.09	mean=1.11 std=0.34 95% CI=±0.008
aromaticity	mean=0.08 std=0.12 95% CI=±0.0097	mean=0.05 std=0.07 95% CI=±0.0049	mean=0.0022 std=0.0075 95% CI=±0.0065	mean=0.0018 std=0.0052 95% CI=±0.0017
instability_index	mean=20.88 std=30.44 95% CI=±2.11	mean=53.8912 std=41.9284 95% CI=±2.32	mean=22.24 std=37.67 95% CI=±3.35	mean=19.87 std=30.29 95% CI=±3.29
molecular_weight	mean=3874.43 std=1015.30 95% CI=±100.90	mean=42781.5 std=4291.7 95% CI=±255.716	mean=8280.03 std=1461.54 95% CI=±129.94	mean=7829.32 std=1392.28 95% CI=±111.3
isoelectric_point	mean=4.50 std=1.32 95% CI=±0.493	mean=10.286 std=2.212 95% CI=±0.03961	mean=8.07 std=3.10 95% CI=±0.27	mean=3.89 std=1.29 95% CI=±0.11

Table 1: Results on 500 sequences of length 100 using the **ESM3** model.

Property	TA	SAE	Fine-Tuned	Base
charge at pH 7	mean=9.085 std=12.15 95% CI=±1.68	mean=1.32 std=1.73 95% CI=±0.05826	mean=0.0090 std=9.010 95% CI=±0.79	mean=-0.4224 std=3.2994 95% CI=±0.4619
hydrophobicity	mean=0.31 std=0.80 95% CI=±0.11	mean=1.65 std=1.28 95% CI=±0.08	mean=-0.20 std=0.91 95% CI=±0.080	mean=-0.1808 std=0.8492 95% CI=±0.1189
aromaticity	mean=0.022 std=0.027 95% CI=±0.0038	mean=0.04 std=0.08 95% CI=±0.005	mean=0.025 std=0.037 95% CI=±0.003	mean=0.0236 std=0.0367 95% CI=±0.0051
instability_index	mean=39.49 std=30.42 95% CI=±4.22	mean=52.4 std=47.3 95% CI=±2.91	mean=100.09 std=76.27 95% CI=±6.72	mean=93.08 std=72.68 95% CI=±6.37
molecular_weight	mean=10685.89 std=642.44 95% CI=±89.04	mean=42635.5 std=4330.7 95% CI=±268.4	mean=9243.45 std=1871.97 95% CI=±164.09	mean=8994.6477 std=2097.8866 95% CI=±290.7522
isoelectric_point	mean=10.79 std=2.23 95% CI=±0.31	mean=9.92 std=2.41 95% CI=±0.15	mean=6.00 std=1.70 95% CI=±0.15	mean=6.0813 std=1.7428 95% CI=±0.2440

Table 2: Results on 500 sequences of length 100 using the **ProGen2-Large** model.

Property	TA	SAE	Fine-Tuned	Base
charge at pH 7	mean=1.54 std=1.37 95% CI=±0.0471	mean=1.88 std=1.42 95% CI=±0.03927	mean=-7.40 std=3.48 95% CI=±0.30	mean=-7.32 std=3.29 95% CI=±0.29
hydrophobicity	mean=1.65 std=1.21 95% CI=±0.28	mean=1.72 std=1.35 95% CI=±0.085	mean=-0.12 std=0.20 95% CI=±0.017	mean=-0.11 std=0.19 95% CI=±0.017
aromaticity	mean=0.031 std=0.076 95% CI=±0.0042	mean=0.045 std=0.98 95% CI=±0.006	mean=0.041 std=0.014 95% CI=±0.001	mean=0.042 std=0.19 95% CI=±0.017
instability_index	mean=39.7 std=44.3 95% CI=±2.55	mean=49.8 std=45.2 95% CI=±2.80	mean=29.86 std=12.09 95% CI=±1.06	mean=30.00 std=11.99 95% CI=±1.05
molecular_weight	mean=41735.7 std=4189.3 95% CI=±259.4	mean=42890.0 std=4400.1 95% CI=±272.0	mean=10540.82 std=226.39 95% CI=±19.84	mean=10550.98 std=229.32 95% CI=±20.10
isoelectric_point	mean=9.45 std=2.20 95% CI=±0.175	mean=9.75 std=2.35 95% CI=±0.145	mean=4.70 std=0.64 95% CI=±0.06	mean=4.74 std=0.61 95% CI=±0.053

Table 3: Results on 500 sequences of length 100 using the **ProLLaMA** model.

B Compute Resources

B.1 Base Model

Base generations for ESM3, ProGen2-Large, and ProLLaMA were performed on the A40. For each model, we generated 500 sequences of length 100 amino acids and computed properties with `ProteinAnalysis`. Wall-clock time varied by model and batch size; no out-of-memory (OOM) events occurred on the A40 configuration.

B.2 Fine-Tuning

Fine-tuning runs (per property and per model) were conducted on the same A40 instance. We used the PLMInterp `train` split without modification and trained small property-specific adapters/checkpoints as described in Section 3.1. After training, we sampled 500 sequences (100 aa) per run and scored the same six properties. Training and evaluation completed reliably on the A40 with default mixed-precision disabled unless otherwise specified.

B.3 Task Arithmetic

TA required one fine-tuned checkpoint per property to construct the task vector and a base checkpoint for application. Both fine-tuning (for vector construction) and steered generation were executed on the A40. Applying the task vector and subsequent sampling did not require additional optimization steps and ran comfortably within A40 memory limits.

B.4 Sparse Autoencoder (SAE)

For the SAE, the training was conducted by using regular GPU for the runtime. Depending on the different property we had to calculate for, that dictates the runtime for the results to display. For this setup specifically, we imported an existing SAE github repo that ran a base model of ESM. Based on the different runtimes, the average runtime was 15-20 minutes.

C Licenses and Attribution

We used the following external models and datasets in our experiments:

- **ESM3.** We use the ESM3 model family as referenced in our paper (Section 3). Accessed via the authors' official distribution/model card; used under the license specified there. We do not redistribute weights.
- **ProGen2-Large** [Nijkamp et al., 2022]. Access via the authors' distribution/model card; used under the license specified by the authors. We do not redistribute weights.
- **ProLLaMA**. Hugging Face model card. Used under the license on the model card. We do not redistribute weights.
- **PLMInterp Dataset** (Protolyze): We used the PLMInterp dataset provided by Parsan et al. [Parsan et al., 2025], available on Hugging Face at `https://huggingface.co/datasets/protolyze/plminterp`. This dataset is provided under the CC BY 4.0 License.
- **Biopython Library**: Used for computing protein properties (GRAVY, charge at pH 7). Licensed under the Biopython License Agreement, compatible with the BSD License. Available at `https://biopython.org/`.
- **SAE Codebase**: The SAE implementation was adapted from `https://github.com/johnyang101/reticular-sae`. Licensed under MIT License.

All external assets were used in accordance with their respective licenses. No modifications were made that would violate redistribution or usage terms.