FOLDSAE: LEARNING TO STEER PROTEIN FOLDING THROUGH SPARSE REPRESENTATIONS

Wojciech Zarzecki
University of Warsaw,
Warsaw University of Technology

Paulina Szymczak *
Helmholtz Munich
paulina.szymczak@helmholtz-munich.de

Ewa Szczurek
University of Warsaw,
Helmholtz Munich

Kamil Deja
Warsaw University of Technology,
IDEAS Research Institute

ABSTRACT

While models like RFdiffusion excel at generating protein backbones, their "black box" nature currently restricts design to a process of stochastic sampling rather than precise engineering. To bridge this gap, we introduce FoldSAE, a framework that adapts Sparse Autoencoders (SAEs) to decompose RFdiffusion's dense activations into interpretable, monosemantic features. We demonstrate that these unsupervised features capture fundamental physical properties, including secondary structure formation and solvent-accessible surface area (SASA). Leveraging these insights, we implement a steering mechanism that enables targeted modulation of backbone folding and surface exposure during the denoising process, both in *de novo* and anlogue generation. Our work pioneers a new framework for making RFdiffusion more interpretable, demonstrating how understanding internal features can be directly translated into precise control over the protein design process.

1 INTRODUCTION

The "black box" nature of deep learning methods presents a significant barrier in their adaptation in life science domains. While state-of-the-art models like RFdiffusion (Watson et al., 2023) show remarkable capabilities in generating novel protein backbones, our inability to understand their internal representations limits scientific insight and practical control. This lack of transparency, means we cannot debug, validate or steer the generative process itself, turning protein design into a matter of sampling and filtering rather then precise engineering.

Mechanistic interpretability aims to solve this issue, by finding human-understandable mechanisms within a model. A promising technique in this area is the Sparse Autoencoder (SAE) Olshausen & Field (1997), which decomposes model's dense **activations** (the vector outputs of network layers during a forward pass) into a sparse set of "mono-semantic" **features** (directions in the activation space corresponding to distinct, interpretable concepts). This approach has provided unprecedented insight into language models (Huben et al., 2024; Bricken et al., 2023; Marks et al., 2025), with some applications in diffusion models (Surkov et al., 2024; Kim et al., 2024; Cywiński & Deja, 2025). In the molecular domain, however, existing work has been restricted to sequence-level representations (Adams et al., 2025; Simon & Zou, 2025; Rives et al., 2021), leaving the potential for interpretable control over structure generation largely unexplored.

In this work, we introduce FoldSAE, a new framework that leverages sparse autoencoders to interpret the protein folding process within RFdiffusion Watson et al. (2023). Our goal is to decompose RFdiffusion's complex, dense representations into a sparse set of monosemantic features, thereby uncovering its inner workings. To validate that these unsupervised features are indeed meaningful and useful, we train a single SAE that we use to analyse (1) the process of secondary structures generation and (2) solvent accessible surface area (SASA) as two complementary proof-of-concept applications. To that end, we first propose a simple heuristics based on block-ablation, to localize the specific parts of the model that are critical for those tasks. We use activations from the most

*Corresponding author. Email: paulina.szymczak@helmholtz-munich.de

relevant block to train a SAE, and use simple linear probing models to identify which of the discovered sparse features correlate with the target outcomes. Our analysis reveals that, even though trained in a fully unsupervised way, the same features often control both helix and strand formation, but with opposite correlation. Crucially, because the SAE is trained without supervision, the same model can be repurposed to identify features predictive of SASA, enabling control over backbone exposure without retraining. This confirms that a single SAE trained on one block captures a rich set of structural properties.

Those observations allow us to demonstrate that interpretability can be directly translated into precise control. We introduce a steering mechanism, where we can amplify or suppress these specific features during the diffusion-denoising process. As a result, we can precisely modulate the final protein structure, for example, by selectively reinforcing the features positively correlated with helices and blocking those correlated negatively, we can increase the amount of helices in generated protein backbones. Similarly, selecting features based on their correlation with SASA allows us to modulate the degree of backbone exposure in the generated designs. FoldSAE thus offers a novel framework that directly links internal model representations to precise control, enabling a more directed protein design process. To facilitate future research, we release our code together with weights of trained SAE models at GitHub.

Our contribution can be summarized as follows: (i) we introduce FoldSAE, a method for training a Sparse Autoencoder using the internal activations of RFdiffusion, successfully decomposing its dense representations into sparse, interpretable features; (ii) we establish a link between specific internal features and protein secondary structure, as well as SASA; and (iii) we design a steering mechanism that allows for precise, tunable control over the secondary structure formation and backbone solvent exposure during the RFdiffusion generative process.

2 Background & Related work

2.1 Protein preliminaries

Protein structure is hierarchically organized: the amino acid sequence dictates local secondary structure elements (helices, strands, and coils) whose spatial packing determines the overall three-dimensional fold and, ultimately, biological function. Computationally, this structure can be reduced to the protein backbone, a repeating chain of three atoms per residue: the amino-group nitrogen (N), the alpha-carbon (C_α), and the carbonyl carbon (C). This representation abstracts away side-chain identity, treating all residues as glycine, with the N-C_α-C coordinates fully specifying each residue's position and orientation in 3D space.

RFdiffusion generates protein backbones by iteratively denoising a rigid-frame representation consisting of C_α coordinates and rotation matrices for each residue. The architecture utilizes 36 stacked blocks that simultaneously refine 1D, 2D, and 3D inputs to predict the necessary translations and rotations for the denoising process. This process yields a backbone structure that requires subsequent processing by models like ProteinMPNN (Dauparas et al., 2022) to assign specific amino acid identities.

2.2 Mechanistic interpretability

Within mechanistic interpretability methods, Sparse Autoencoders (SAEs) (Olshausen & Field, 1997) have emerged as a powerful tool to disentangle dense representations, into steerable features. This approach, specifically with top-k sparsity Makhzani & Frey (2013), have been successfully applied to Large Language Models (Huben et al., 2024; Bricken et al., 2023; Marks et al., 2025), while recently, works by Surkov et al. (2024), Kim et al. (2024), and Cywiński & Deja (2025) have also demonstrated that the same tool can be used for precise control in image diffusion models. In biology, the utility of SAE has not been fully discovered. While works like InterProt (Adams et al., 2025) and InterPLM (Simon & Zou, 2025) utilize SAEs on ESM2 embeddings (Rives et al., 2021) for sequence design, they leave structure generation unexplored (Garcia & Ansuini, 2025), a gap this work fills by introducing interpretable steering to RFdiffusion.

3 Method

Our method aims to find features within the model activations, which encode information about interpretable properties of protein backbones to use them for steering during the inference process. It consists of three stages: **localization**, **interpretation** and **intervention**. In the stage of localization,

we identify the crucial block encoding semantic information about protein backbone design. In the interpretation stage, we decompose activations of the chosen block and select those corresponding to the interesting target. Finally, in the intervention stage, we manipulate the identified features to steer the generation towards desired properties.

3.1 LOCALIZATION

The first crucial design choice is the selection of the block to intervene. To that end, we adapt the method introduced by Cywiński & Deja (2025), and determine which block encodes information about desired properties, by performing iterative ablation of the blocks. We understand ablation of n^{th} block as substituting its output with an output of the previous - n^{th-1} block. As shown by Cywiński & Deja (2025), if a given block adds information about desired properties, its ablation should result in observable distribution shift in the final generations. Formally, let S denote a metric quantifying the target property for a given model configuration. We identify the optimal block index m^* by finding the ablation that maximizes the metric deviation: $m^* = \text{argmax}_m |S(\mathcal{M}_{\text{orig}}) - S(\mathcal{M}_{\setminus m})|$, where $\mathcal{M}_{\text{orig}}$ denotes the original model and $\mathcal{M}_{\setminus m}$ the model with the m-th block ablated.

3.2 INTERPRETATION

Given the activations from the selected block, our goal is to decompose them into interpretable features by training a SAE.

We train a Top-K SAE to reconstruct activations on a per-residue basis. We flatten the block outputs into l sequential segments, treating each residue's embedding as an independent patch $\mathbf{x} \in \mathbb{R}^d$. The model consists of a single-layer encoder that encode activations x into features: $\mathbf{z} = \text{TopK}(\text{ReLU}(\mathbf{W}_{\text{enc}}(\mathbf{x} - \mathbf{b})))$ and decoder that reconstructs them as $\hat{\mathbf{x}} = \mathbf{W}_{\text{dec}}\mathbf{z} + \mathbf{b}$, where $\mathbf{W}_{\text{enc}} \in \mathbb{R}^{n \times d}$ and $\mathbf{W}_{\text{dec}} \in \mathbb{R}^{d \times n}$ are weight matrices, and $\mathbf{b} \in \mathbb{R}^d$ is a learnable bias. The TopK operation enforces sparsity by zeroing out all but the k highest latent values. The size of the latent dimension n is a result of scaling d by a fixed *expansion factor*. We optimize the model by minimizing the reconstruction error objective: $\mathcal{L}(\mathbf{x}) = \|\mathbf{x} - \hat{\mathbf{x}}\|_2^2$. A well-trained SAE effectively decomposes dense activations into a dictionary of n sparse, monosemantic feature vectors (columns of $\mathbf{W}_{\text{dec}}$), which can be directly manipulated to steer the generative process.

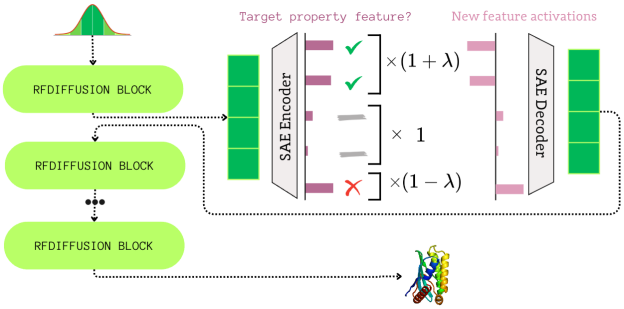


Figure 1: Overview of the FoldSAE steering mechanism. During the protein backbone generation process, activations from the localized RFdiffusion block are intercepted and decomposed into sparse features by the SAE Encoder. These features are then modulated based on their correlation with the desired target property (identified via probing classifiers). To steer the trajectory, features positively correlated with the target are amplified by a factor of $(1 + \lambda)$, while negatively correlated features are suppressed by $(1 - \lambda)$; neutral features remain unmodified (scaled by 1). The adjusted features are reconstructed by the SAE Decoder and reintroduced into the network to guide subsequent diffusion steps.

3.3 INTERVENTION

Let us assume, that we have identified a set of interesting SAE features that correlate (positively or negatively) with the desired property. We can use those features to steer the generation process through interventions by passing all of the activations through the autoencoder, while suppressing features negatively correlated with the target property, and reinforcing the positively correlated ones, leaving all others unchanged. We introduce a hyper-parameter, λ, that controls both the direction and magnitude of the intervention: $\lambda = 0$ corresponds to no modification, $\lambda > 0$ steers toward the target property, and $\lambda < 0$ steers away from it. Concretely, each SAE feature is scaled by $(1 + \lambda)$ if positively correlated with the target, $(1 - \lambda)$ if negatively correlated, and left unchanged otherwise (see Figure 1).

4 EXPERIMENTS

4.1 LOCALIZATION

To identify the specific RFdiffusion block responsible for encoding our target properties–secondary structure and SASA–we conduct a systematic ablation study as outlined in Section 3.1. Secondary structure is annotated using STRIDE Frishman & Argos (1995) with the standard eight-to-three-state reduction Rost & Sander (1993), while SASA is computed via the Shrake–Rupley algorithm as implemented in FreeSASA Mitternacht (2016) (see Appendix C). By iteratively removing blocks and evaluating the generated structures, we observe that the ablation of block *main_04* yields the most profound impact. Specifically, it renders the model incapable of generating alpha-helices and induces the largest deviation in SASA distributions compared to the baseline (see Appendix, Figure 10 and Figure 11). Consequently, we select block *main_04* for all subsequent interpretation and intervention experiments.

4.2 INTERPRETATION

Feature selection Using the single SAE trained in the previous step, we identify features that discriminate between classes of our target properties by fitting logistic regression models to the fixed SAE's latent features to predict the corresponding property class. We describe details of dataset gathering and models training in Appendix F. In our experiments, we identify features that discriminate between binary target classes. For secondary structures, we utilize 'helix vs. others' and 'strand vs. others' classifiers; for SASA, we establish low and high categories based on the 25th and 75th percentiles of observed values. We select indices where both models' coefficients exceed the threshold and have **opposite signs**. For example, a feature that is strongly positive for the helix classifier and strongly negative for the strand classifier is highly discriminative between them.

4.3 INTERVENTION

To evaluate whether the discovered features are causxnally linked to generation outcomes, we perform a series of interventions to steer the process toward specific secondary structures or surface properties. We vary the steering intensity λ across empirically determined ranges. To ensure precision, prior to intervention, we employ pre-trained linear regressors used for features selection, to assess whether the activation patch for a specific residue already exhibits the target property (e.g., whether it is classified as a helix when steering towards helices); we proceed with intervention only if the target property is absent.

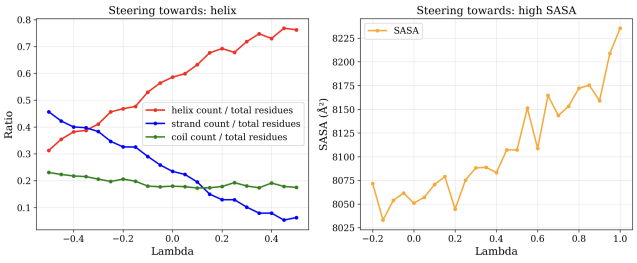
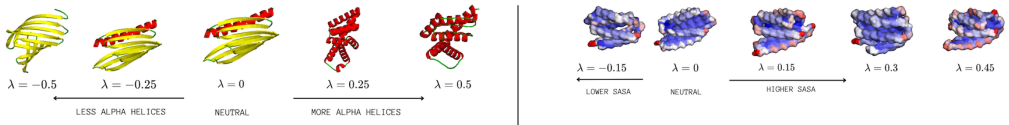


Figure 2: FoldSAE intervention results. (Left) Fraction of residues assigned to helices (red), strands (blue), and coils (green) as a function of steering intensity λ when steering towards helices. (Right) Solvent Accessible Surface Area (SASA) of generated backbones as a function of λ when steering towards high-SASA features, showing a clear ability to modulate surface exposure.

Subsequently, we analyze the distribution of helices, strands, and coils, as well as the SASA within the generated protein backbones. When steering toward specific secondary structures, increasing λ to positive values monotonically elevates the proportion of the target structure while reducing the others (Figure 2, left). We extend this framework to physical properties by steering towards features associated with high solvent exposure. As illustrated in Figure 2 (right), we observe a direct correlation between λ and the mean SASA of the generated backbones, demonstrating that FoldSAE can modulate not just local geometry but other protein properties. Notably, steering in order to increase exposure is more effective than toward decreasing it. This is consistent with the model's training distribution of natively folded, compact structures that already exhibit relatively low solvent accessibility.

Moreover, we observe fine-grained control over the intervention even at the level of individual protein backbones. As shown in Figure 3, increasing λ visibly alters the structural composition, re-



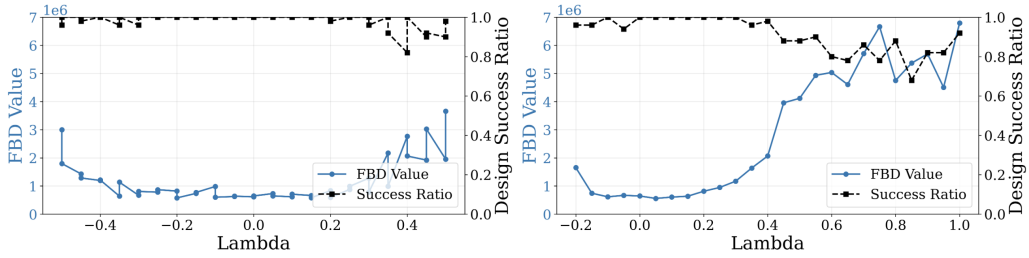
(a) Steering secondary structure ($\lambda \in [-0.5, 0.5]$). **(b)** Steering SASA ($\lambda \in [-0.15, 0.45]$).

Figure 3: Qualitative Analysis of Targeted Steering. The figure displays generated backbones under varying steering intensities. Steering towards helices results in progressively more helical content (red ribbons). **(b)** Steering towards high SASA results in less compact, more exposed structures (visualized by surface potential).

sulting in a higher density of helices or a more expanded, accessible surface area depending on the steering target. More designs can be found in Appendix, Figure 5.

Validation of generated structures To assess the biological plausibility of generated protein backbones following intervention, we compare their distribution against backbones generated without intervention. We embed backbones using ESM3 (Hayes et al., 2025) and quantify distributional alignment using FBD (Møller-Larsen et al., 2025) (an adaptation of FID (Heusel et al., 2017) for proteins). To fairly evaluate structural quality independent of topological shifts, we weight reference proteins so that their distribution of helix-to-strand ratios (for secondary structure interventions) or SASA values (for SASA interventions) matches that of the corresponding generated batch. This ensures that any observed quality differences stem from the steering mechanism itself rather than from shifts in the overall structural composition.

We independently evaluate batches of protein backbones across varying intervention strengths λ; the results are detailed in Appendix, Figure 4. We observe that FBD scores for secondary structure interventions remain close to the neutral baseline ($\lambda = 0$) across most of the steering range. This indicates that our mechanism preserves structural integrity and produces backbones resembling natural reference distributions. At higher intensities ($|\lambda| > 0.4$), the design success rate drops and FBD increases, aligning with the structural collapse observed in qualitative samples. For SASA interventions, the effective range of λ where backbones remain intact is narrower, as extreme surface constraints may push the model toward biologically infeasible configurations.



(a) Secondary structure interventions. **(b)** SASA interventions.

Figure 4: Biological plausibility of steered backbones. The Fréchet Biological Distance (FBD) scores (solid lines) and design success rates (dashed lines) are shown across a range of intervention strengths (λ) for both (a) secondary structure and (b) SASA interventions. FBD scores remain stable and close to the neutral baseline ($\lambda = 0$) for moderate steering intensities, confirming that FoldSAE generates biologically realistic structures before extreme constraints induce structural collapse.

5 Conclusions

In this work, we introduce FoldSAE, a framework leveraging Sparse Autoencoders to decompose RFdiffusion's internal representations in a fully unsupervised manner. To demonstrate that discovered features are biologically meaningful and interpretable, we have shown that we can use SAE features selected from a single model to steer the generative process towards desired secondary structure formation or solvent accessible surface area (SASA). We note that the current analysis operates on protein backbones, and SASA estimates would benefit considerably from full atomic detail including side chains; however, the proposed framework is model-agnostic and readily extends to all-atom generative models as they become available.

MEANINGFULNESS STATEMENT

By decomposing learned representations into sparse, interpretable features and enabling precise structural control, FoldSAE contributes to fundamental understanding of generative models for biology, with implications for reliable, interpretable design pipelines in clinical protein engineering.

ACKNOWLEDGMENTS

This work was funded by the National Science Centre, Poland, grant no UMO-2023/51/B/ST6/03004 and the European Research Council (ERC) under the European Funding Union's Horizon 2020 research and innovation programme (grant agreement No 810115 – DOG-AMP). The computing resources were provided by the PL-Grid Infrastructure, grant no.: PLG/2025/18390 and PLG/2025/18391.

REFERENCES

Etowah Adams, Liam Bai, Minji Lee, Yiyang Yu, and Mohammed AlQuraishi. From mechanistic interpretability to mechanistic biology: Training, evaluating, and interpreting sparse autoencoders on protein language models. *bioRxiv*, 2025.

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Zac Hatfield-Dodds, Alex Tamkin, Karina Nguyen, Brayden McLean, Josiah E Burke, Tristan Hume, Shan Carter, Tom Henighan, and Christopher Olah. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2023. https://transformer-circuits.pub/2023/monosemantic-features/index.html.

Bartosz Cywiński and Kamil Deja. Saeuron: Interpretable concept unlearning in diffusion models with sparse autoencoders. *arXiv preprint arXiv:2501.18052*, 2025.

Jonas Dauparas, Ivan Anishchenko, Nathan Bennett, Hanyu Bai, Robert J Ragotte, Lukas F Milles, Basile IM Wicky, Alexis Courbet, Roel J de Haas, Noah Bethel, Peter JY Leung, Timothy F Huddy, Sarel Pellock, David Tischer, Frank Chan, Brad Koepnick, Huyen Nguyen, Ahyeon Kang, Akash K Bera, Neil P King, and David Baker. Robust deep learning–based protein sequence design using ProteinMPNN. *Science*, 378(6615):49–56, 2022. doi: 10.1126/science.add2187.

Dmitrij Frishman and Patrick Argos. Knowledge-based protein secondary structure assignment. *Proteins: Structure, Function, and Genetics*, 23(4):566–579, 1995.

Edith Natalia Villegas Garcia and Alessio Ansuini. Interpreting and steering protein language models through sparse autoencoders. *arXiv preprint arXiv:2502.09135*, 2025.

Thomas Hayes, Roshan Rao, Halil Akin, Nicholas J. Sofroniew, Deniz Oktay, Zeming Lin, Robert Verkuil, Vincent Q. Tran, Jonathan Deaton, Marius Wiggert, Rohil Badkundri, Irhum Shafkat, Jun Gong, Alexander Derry, Raul S. Molina, Neil Thomas, Yousuf A. Khan, Chetan Mishra, Carolyn Kim, Liam J. Bartie, Matthew Nemeth, Patrick D. Hsu, Tom Sercu, Salvatore Candido, and Alexander Rives. Simulating 500 million years of evolution with a language model. *Science*, 2025. doi: 10.1126/science.ads0018. URL `http://dx.doi.org/10.1126/science.ads0018`.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

Robert Huben, Hoagy Cunningham, Logan Riggs Smith, Aidan Ewart, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=F76bwRSLeK`.

Dahye Kim, Xavier Thomas, and Deepti Ghadiyaram. *Revelio*: Interpreting and leveraging semantic information in diffusion models. *arXiv preprint arXiv:2411.16725*, 2024.

Alireza Makhzani and Brendan J. Frey. k-sparse autoencoders. *CoRR*, abs/1312.5663, 2013. URL https://api.semanticscholar.org/CorpusID:14850799.

Samuel Marks, Can Rager, Eric J Michaud, Yonatan Belinkov, David Bau, and Aaron Mueller. Sparse feature circuits: Discovering and editing interpretable causal graphs in language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=I4e82CIDxv.

Simon Mitternacht. Freesasa: An open source c library for solvent accessible surface area calculations. *F1000Research*, 5:189, February 2016. ISSN 2046-1402. doi: 10.12688/f1000research.7931.1. URL http://dx.doi.org/10.12688/f1000research.7931.1.

Rasmus Møller-Larsen, Adam Izdebski, Jan Olszewski, Pankhil Gawade, Michal Kmicikiewicz, Wojciech Zarzecki, and Ewa Szczurek. seqme: a Python library for evaluating biological sequence design, nov 2025. URL https://arxiv.org/abs/2511.04239.

Bruno A. Olshausen and David J. Field. Sparse coding with an overcomplete basis set: A strategy employed by v1? *Vision Research*, 37:3311–3325, 1997. URL https://api.semanticscholar.org/CorpusID:14208692.

Alexander Rives, Joshua Meier, Tom Sercu, Siddharth Goyal, Zeming Lin, Jason Liu, Demi Guo, Myle Ott, C Lawrence Zitnick, Jerry Ma, et al. 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.

Burkhard Rost and Chris Sander. Prediction of protein secondary structure at better than 70% accuracy. *Journal of molecular biology*, 232(2):584–599, 1993.

Elana Simon and James Zou. Interplm: Discovering interpretable features in protein language models via sparse autoencoders. *Nature Methods*, pp. 1–11, 2025.

Viacheslav Surkov, Chris Wendler, Mikhail Terekhov, Justin Deschenaux, Robert West, and Caglar Gulcehre. Unpacking sdxl turbo: Interpreting text-to-image models with sparse autoencoders. *arXiv preprint arXiv:2410.22366*, 2024.

Joseph L Watson, David Juergens, Nathaniel R Bennett, Brian L Trippe, Jason Yim, Helen E Eisenach, Woody Ahern, Andrew J Borst, Robert J Ragotte, M. K. K., et al. De novo design of protein structure and function with rfdiffusion. *Nature*, 620(7976):1090–1100, 2023. doi: 10.1038/s41586-023-06415-8. URL https://www.nature.com/articles/s41586-023-06415-8.

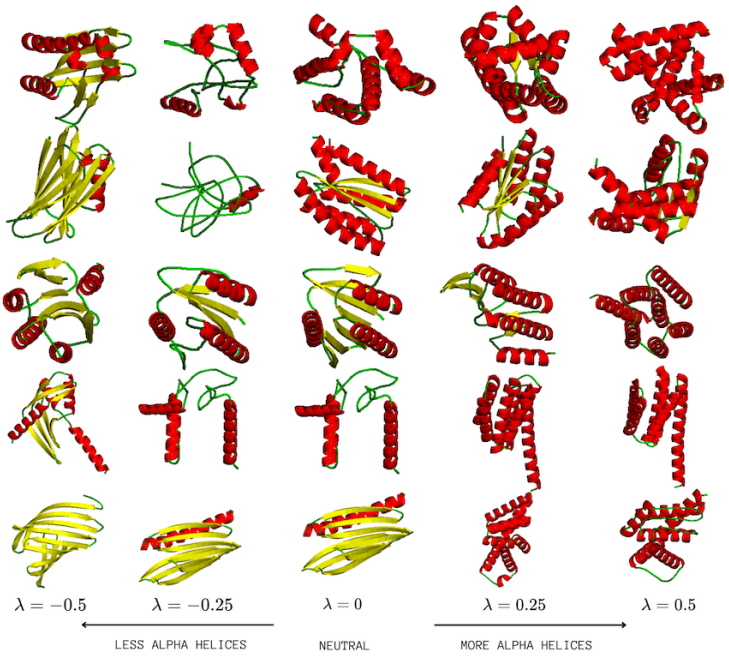
A Appendix

B Results of intervention

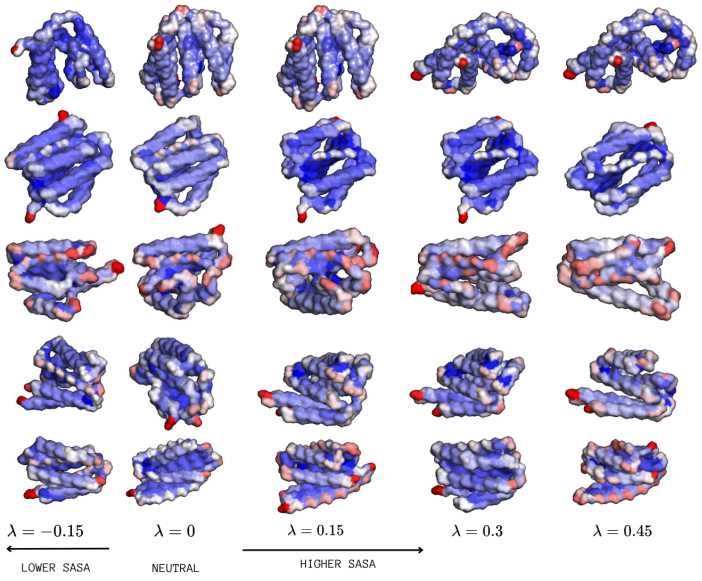
Figure 5 shows visualisation of targeted steering for multiple backbones. Figure 6 shows quantitive results for steering towards helix, strand and high SASA side by side.

C Evaluated properties

While the proposed methodology is general and allows for unsupervised discovery of interpretable features, in this section, we propose to validate whether Sparse Autoencoder learns features that allow for differentiation between the final secondary structure of the generated backbone. Additionally, we evaluate solvent-accessible surface area (SASA) as a complementary, physically grounded measure of residue exposure that can be computed directly from generated structures.



(a) Backbone generation under varying steering intensities. Increased intensity correlates with higher helical content.



(b) Backbone generation under varying steering intensities. Increased intensity correlates with brighter content.

Figure 5: Visual analysis of generated structures: (a) illustrates the effect of steering on secondary structure distribution, while (b) shows the resulting surface properties/SASA.

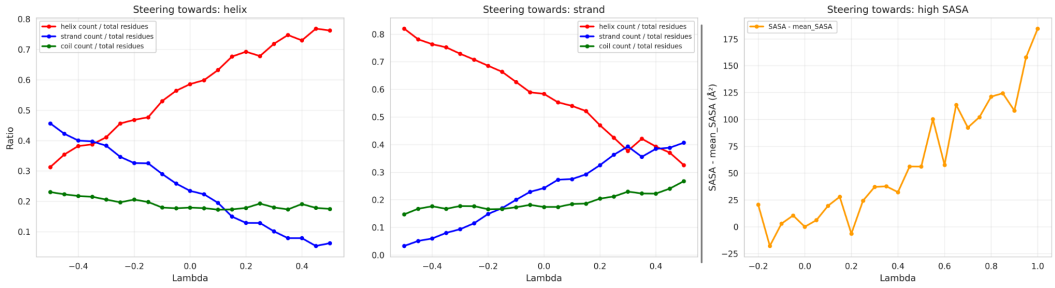


Figure 6: FoldSAE intervention results. (Left and Middle) Fraction of residues assigned to helices (red), strands (blue), and coils (green) as a function of steering intensity λ when steering towards helices and strands, respectively. (Right) Solvent Accessible Surface Area (SASA) of generated backbones as a function of λ when steering towards high-SASA features, showing a clear ability to modulate surface exposure.

Secondary structure To annotate the generated backbones, we use STRIDE Frishman & Argos (1995), which assigns secondary structure classes to each residue based on hydrogen-bond energetics and torsion-angle propensities. We reduce the eight-state assignments to three states by mapping helical conformations (H, G, I) to helix, extended conformations (E, B) to strand, and all remaining states to coil, following the standard reduction scheme Rost & Sander (1993). For evaluation, we measure ratio of given class to all residues.

We examine how frequent are helix, strand and coil in protein backbones generated in unconditional manner, as shown in Figure 7.

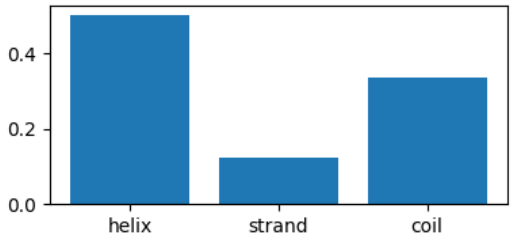


Figure 7: Distribution of helices, strands and coils in training dataset. We observe predominance of helices what matches distribtuion of secondary structures in natural proteins.

SASA We quantify residue exposure using solvent-accessible surface area (SASA), which measures the surface area of a biomolecule accessible to a solvent probe. We compute SASA using the Shrake-Rupley algorithm with FreeSASA Mitternacht (2016), an open-source implementation of standard SASA calculation routines. Since our method operates on generated backbones, we report backbone SASA computed from the atoms present in the structure and aggregate it across residues. When analyzing the relationship between SAE features and SASA, we operate on per-residue SASA.

We examine distribution of SASA in training datasets in Figure 8.

D SAE IN RFDIFFUSION

SAE training We conducted a hyperparameter grid search for SAE training, sweeping across learning rates, expansion factors, and the sparsity parameter k; full details are provided in Appendix (see Table 1). Our final SAE was trained for 50,000 steps with a batch size of 4,096, employing a learning rate of 1×10^{-4}, an expansion factor of 16, and $k = 64$. This configuration achieves an explained variance of 99.1%, while maintaining a low fraction of both dead features (defined as latent neurons activating on fewer than 1 in 10^6 training samples) and high-frequency features (activating on more than 1 in 100 examples). Minimizing the latter is crucial, as frequent features

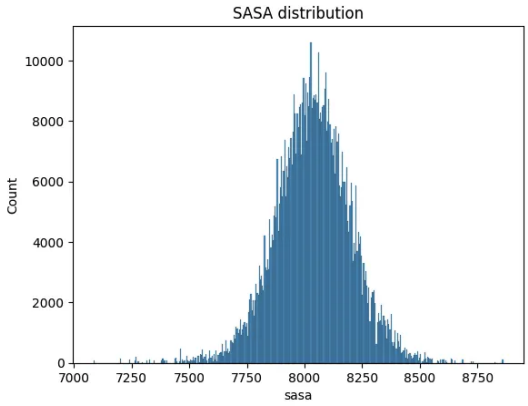


Figure 8: Distribution of SASA of proteins in training dataset. We observe normal distribution what support realism of generated backbones.

are prone to encoding multiple properties rather than being mono-semantic. The visualization of feature density is shown in Appendix, Figure 9.

We gather a dataset for SAE training by collecting activations from chosen block for a set of 1200 protein backbones generated without any conditioning, for each timestep of diffusion process. To operate on single residue level, we flatten each collected activations vector and split it into l patches, where l denotes the number of residues in the protein. Then as described in Section 3.2 we train SAE to reconstruct activation patch for single residue. Table 1 presents the full scope of our grid search across learning rates, expansion factors, and k values from grid search described in Section D. Figure 9 illustrates the distribution of feature activations in the final model, confirming a desirable sparsity profile with minimal dead or poly-semantic features.

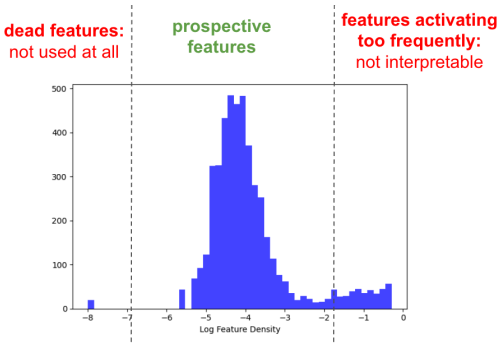


Figure 9: Distribution of log feature density for the trained SAE. The histogram illustrates the frequency of feature activations. The training setup results in a desirable distribution with a minimal fraction of dead features (left tail, $< 10^{-6}$) and high-frequency poly-semantic features (right tail, $> 10^{-2}$).

SAE at intervention When intervention is required, we must address the fact that reconstructing activations with SAE inherently introduces reconstruction error. If propagated to subsequent blocks, this error causes a distribution shift in activations that can degrade downstream performance. To mitigate this, we offset the error using the following procedure:

1. We reconstruct the original activations Γ without applying any intervention, yielding $\hat{\Gamma}$.
2. We then calculate the original reconstruction error as $E = \hat{\Gamma} - \Gamma$.
3. We reconstruct the activations Γ *while applying* the intervention, which results in $\hat{\Gamma}'$.
4. The final activations returned after the intervention are the intervened reconstruction offset by the original error: $\hat{\Gamma}' + E$.

Table 1: SAE training grid search. We train SAE for various expansion factors (how many times latent space is wider), k in TopK, learning rate and report explained variance (the higher the better), ratio of dense features (sparsity below 10^{-2}) (the lower the better) and ration of alive neuons (with sparsity below 10^{-5}) (the higher the better).

Expl. Var.	Spars. $< 10^{-5}$	Spars. $< 10^{-2}$	Exp.	LR	k	Expl. Var.	Spars. $< 10^{-5}$	Spars. $< 10^{-2}$	Exp.	LR	k	Expl. Var.	Spars. $< 10^{-5}$	Spars. $< 10^{-2}$	Exp.	LR	k	Expl. Var.	Spars. $< 10^{-5}$	Spars. $< 10^{-2}$	Exp.	LR	k
0.81	0.00e+00	0.47	32	0.0000	512	1.00	7.32e-01	0.88	8	0.0050	128	1.00	1.68e-01	0.84	32	0.0005	512	1.00	6.61e-01	0.87	8	0.0010	128
1.00	8.46e-02	0.81	32	0.0005	512	1.00	7.37e-01	0.84	8	0.0010	128	0.99	0.00e+00	0.66	32	0.0001	512	0.75	8.53e-02	0.72	8	0.0000	64
1.00	0.00e+00	0.67	32	0.0001	512	0.89	8.36e-02	0.79	8	0.0000	64	1.00	0.00e+00	0.00	32	0.0050	512	1.00	5.65e-01	0.93	8	0.0005	64
1.00	7.17e-01	0.89	32	0.0050	512	1.00	6.93e-01	0.89	8	0.0005	64	1.00	2.85e-01	0.86	32	0.0010	512	0.99	8.02e-03	0.84	8	0.0001	64
1.00	2.83e-01	0.84	32	0.0010	512	0.99	1.44e-02	0.84	8	0.0001	64	0.68	2.53e-03	0.64	32	0.0000	256	1.00	8.85e-01	0.95	8	0.0050	64
0.88	6.33e-04	0.72	32	0.0000	256	1.00	9.03e-01	0.94	8	0.0050	64	1.00	3.79e-01	0.92	32	0.0005	256	1.00	7.47e-01	0.94	8	0.0010	64
1.00	4.14e-01	0.89	32	0.0005	256	1.00	8.53e-01	0.92	8	0.0010	64	0.99	0.00e+00	0.81	32	0.0001	256	0.62	2.77e-01	0.90	8	0.0000	32
1.00	0.00e+00	0.84	32	0.0001	256	0.80	4.28e-01	0.87	8	0.0000	32	1.00	0.00e+00	0.94	32	0.0050	256	1.00	6.25e-01	0.97	8	0.0005	32
0.99	7.23e-01	0.93	32	0.0050	256	1.00	8.22e-01	0.94	8	0.0005	32	1.00	5.96e-01	0.93	32	0.0010	256	0.98	1.86e-01	0.94	8	0.0001	32
1.00	6.84e-01	0.91	32	0.0010	256	0.98	2.63e-01	0.92	8	0.0001	32	0.66	4.11e-02	0.79	32	0.0000	128	1.00	8.79e-01	0.97	8	0.0050	32
0.90	1.02e-02	0.85	32	0.0000	128	1.00	9.40e-01	0.96	8	0.0050	32	1.00	5.82e-01	0.96	32	0.0005	128	1.00	8.07e-01	0.97	8	0.0010	32
1.00	7.38e-01	0.94	32	0.0005	128	1.00	8.96e-01	0.95	8	0.0010	32	0.99	1.06e-04	0.88	32	0.0001	128	0.98	0.00e+00	0.00	4	0.0000	512
0.99	1.58e-03	0.92	32	0.0001	128	0.98	0.00e+00	0.00	4	0.0000	512	0.99	0.00e+00	0.97	32	0.0050	128	1.00	0.00e+00	0.05	4	0.0005	512
1.00	9.55e-01	0.97	32	0.0050	128	1.00	0.00e+00	0.02	4	0.0005	512	1.00	8.05e-01	0.97	32	0.0010	128	1.00	0.00e+00	0.00	4	0.0001	512
1.00	9.23e-01	0.96	32	0.0010	128	1.00	0.00e+00	0.00	4	0.0001	512	0.63	2.12e-01	0.90	32	0.0000	64	1.00	0.00e+00	0.00	4	0.0050	512
0.90	1.60e-01	0.94	32	0.0000	64	1.00	0.00e+00	0.06	4	0.0050	512	1.00	7.49e-01	0.98	32	0.0005	64	1.00	8.45e-04	0.06	4	0.0010	512
1.00	8.67e-01	0.97	32	0.0005	64	1.00	0.00e+00	0.06	4	0.0010	512	0.99	1.63e-02	0.95	32	0.0001	64	0.93	0.00e+00	0.02	4	0.0000	256
0.99	7.07e-02	0.96	32	0.0001	64	0.96	0.00e+00	0.02	4	0.0000	256	1.00	9.38e-01	0.99	32	0.0050	64	1.00	2.79e-02	0.49	4	0.0005	256
1.00	9.73e-01	0.98	32	0.0050	64	1.00	3.38e-03	0.46	4	0.0005	256	1.00	9.12e-01	0.98	32	0.0010	64	0.99	0.00e+00	0.15	4	0.0001	256
1.00	9.44e-01	0.98	32	0.0010	64	0.99	0.00e+00	0.15	4	0.0001	256	0.58	4.11e-01	0.97	32	0.0000	32	1.00	4.39e-02	0.54	4	0.0050	256
0.82	6.83e-01	0.97	32	0.0000	32	1.00	4.22e-02	0.53	4	0.0050	256	1.00	8.63e-01	0.99	32	0.0005	32	1.00	2.12e-01	0.52	4	0.0010	256
1.00	9.30e-01	0.98	32	0.0005	32	1.00	2.20e-01	0.51	4	0.0010	256	0.98	2.52e-01	0.98	32	0.0001	32	0.87	0.00e+00	0.21	4	0.0000	128
0.98	6.25e-01	0.98	32	0.0001	32	0.93	0.00e+00	0.29	4	0.0000	128	1.00	9.72e-01	0.99	32	0.0050	32	1.00	2.10e-01	0.76	4	0.0005	128
1.00	9.82e-01	0.99	32	0.0050	32	1.00	1.41e-01	0.71	4	0.0005	128	1.00	9.46e-01	0.99	32	0.0010	32	0.99	0.00e+00	0.41	4	0.0001	128
1.00	9.67e-01	0.99	32	0.0010	32	0.99	0.00e+00	0.52	4	0.0001	128	0.83	0.00e+00	0.19	16	0.0000	512	1.00	2.71e-01	0.76	4	0.0050	128
0.88	0.00e+00	0.19	16	0.0000	512	1.00	6.17e-01	0.77	4	0.0050	128	1.00	7.09e-02	0.71	16	0.0005	512	1.00	4.25e-01	0.76	4	0.0010	128
1.00	2.13e-02	0.68	16	0.0005	512	1.00	4.18e-01	0.74	4	0.0010	128	0.99	0.00e+00	0.46	16	0.0001	512	0.76	4.31e-02	0.53	4	0.0000	64
0.99	0.00e+00	0.42	16	0.0001	512	0.88	3.97e-02	0.60	4	0.0000	64	1.00	0.00e+00	0.00	16	0.0050	512	1.00	4.44e-01	0.87	4	0.0005	64
1.00	1.72e-01	0.77	16	0.0050	512	1.00	5.44e-01	0.81	4	0.0005	64	1.00	1.76e-01	0.74	16	0.0010	512	0.99	8.45e-04	0.73	4	0.0001	64
1.00	1.07e-01	0.71	16	0.0010	512	0.99	8.45e-04	0.73	4	0.0001	64	0.80	1.27e-03	0.44	16	0.0000	256	1.00	7.45e-01	0.89	4	0.0050	64
0.92	0.00e+00	0.51	16	0.0000	256	1.00	7.86e-01	0.88	4	0.0050	64	1.00	2.92e-01	0.86	16	0.0005	256	1.00	5.98e-01	0.89	4	0.0010	64
1.00	2.71e-01	0.81	16	0.0005	256	1.00	7.32e-01	0.84	4	0.0010	64	0.99	0.00e+00	0.68	16	0.0001	256	0.61	1.79e-01	0.80	4	0.0000	32
1.00	0.00e+00	0.71	16	0.0001	256	0.79	2.68e-01	0.74	4	0.0000	32	1.00	1.82e-01	0.88	16	0.0050	256	1.00	4.84e-01	0.93	4	0.0005	32
1.00	6.14e-01	0.88	16	0.0050	256	1.00	7.09e-01	0.89	4	0.0005	32	1.00	6.50e-01	0.88	16	0.0010	256	0.98	1.01e-01	0.90	4	0.0001	32
1.00	6.21e-01	0.83	16	0.0010	256	0.98	1.11e-01	0.85	4	0.0001	32	0.75	2.77e-02	0.66	16	0.0000	128	1.00	7.09e-01	0.94	4	0.0050	32
0.92	7.81e-03	0.72	16	0.0000	128	1.00	8.90e-01	0.93	4	0.0050	32	1.00	5.29e-01	0.93	16	0.0005	128	1.00	6.22e-01	0.94	4	0.0010	32
1.00	6.91e-01	0.89	16	0.0005	128	1.00	8.36e-01	0.91	4	0.0010	32	0.99	0.00e+00	0.80	16	0.0001	128	0.96	0.00e+00	0.00	2	0.0000	512
0.99	4.22e-04	0.85	16	0.0001	128	0.96	0.00e+00	0.00	2	0.0000	512	1.00	8.22e-01	0.95	16	0.0050	128	1.00	0.00e+00	0.00	2	0.0005	512
1.00	8.86e-01	0.94	16	0.0050	128	1.00	0.00e+00	0.00	2	0.0005	512	1.00	8.06e-01	0.94	16	0.0010	128	0.99	0.00e+00	0.00	2	0.0001	512
1.00	8.48e-01	0.91	16	0.0010	128	0.99	0.00e+00	0.00	2	0.0001	512	0.71	1.43e-01	0.82	16	0.0000	64	1.00	0.00e+00	0.00	2	0.0050	512
0.89	1.26e-01	0.89	16	0.0000	64	1.00	0.00e+00	0.00	2	0.0050	512	1.00	6.52e-01	0.97	16	0.0005	64	1.00	0.00e+00	0.00	2	0.0010	512
1.00	8.68e-01	0.95	16	0.0005	64	1.00	0.00e+00	0.00	2	0.0010	512	0.99	6.55e-03	0.89	16	0.0001	64	0.95	0.00e+00	0.00	2	0.0000	256
0.99	7.35e-02	0.91	16	0.0001	64	0.96	0.00e+00	0.00	2	0.0000	256	1.00	9.16e-01	0.97	16	0.0050	64	1.00	0.00e+00	0.02	2	0.0005	256
1.00	9.51e-01	0.97	16	0.0050	64	1.00	0.00e+00	0.02	2	0.0005	256	1.00	8.63e-01	0.97	16	0.0010	64	0.99	0.00e+00	0.00	2	0.0001	256
1.00	9.21e-01	0.96	16	0.0010	64	0.99	0.00e+00	0.00	2	0.0001	256	0.61	3.89e-01	0.94	16	0.0000	32	1.00	0.00e+00	0.07	2	0.0050	256
0.81	5.84e-01	0.94	16	0.0000	32	1.00	0.00e+00	0.05	2	0.0050	256	1.00	8.53e-01	0.98	16	0.0005	32	1.00	0.00e+00	0.05	2	0.0010	256
1.00	8.78e-01	0.97	16	0.0005	32	1.00	0.00e+00	0.03	2	0.0010	256	0.98	2.69e-01	0.97	16	0.0001	32	0.87	1.69e-03	0.04	2	0.0000	128
0.98	4.52e-01	0.96	16	0.0001	32	0.92	0.00e+00	0.05	2	0.0000	128	1.00	9.44e-01	0.98	16	0.0050	32	1.00	2.36e-02	0.48	2	0.0005	128
1.00	9.71e-01	0.98	16	0.0050	32	1.00	4.39e-02	0.45	2	0.0005	128	1.00	8.95e-01	0.98	16	0.0010	32	0.99	0.00e+00	0.16	2	0.0001	128
1.00	9.27e-01	0.97	16	0.0010	32	0.99	0.00e+00	0.16	2	0.0001	128	0.93	0.00e+00	0.02	8	0.0000	512	1.00	5.41e-02	0.53	2	0.0050	128
0.95	0.00e+00	0.02	8	0.0000	512	1.00	8.61e-02	0.56	2	0.0050	128	1.00	2.11e-03	0.49	8	0.0005	512	1.00	1.45e-01	0.52	2	0.0010	128
1.00	4.22e-04	0.44	8	0.0005	512	1.00	2.20e-01	0.49	2	0.0010	128	0.99	0.00e+00	0.20	8	0.0001	512	0.74	1.69e-02	0.23	2	0.0000	64
1.00	0.00e+00	0.16	8	0.0001	512	0.86	1.18e-02	0.25	2	0.0000	64	1.00	0.00e+00	0.00	8	0.0050	512	1.00	1.82e-01	0.75	2	0.0005	64
1.00	3.68e-01	0.57	8	0.0050	512	1.00	2.92e-01	0.71	2	0.0005	64	1.00	7.60e-02	0.51	8	0.0010	512	0.98	1.69e-03	0.51	2	0.0001	64
1.00	1.31e-02	0.48	8	0.0010	512	0.99	0.00e+00	0.53	2	0.0001	64	0.87	0.00e+00	0.21	8	0.0000	256	1.00	3.97e-01	0.77	2	0.0050	64
0.94	0.00e+00	0.25	8	0.0000	256	1.00	5.42e-01	0.76	2	0.0050	64	1.00	2.55e-01	0.75	8	0.0005	256	1.00	3.50e-01	0.76	2	0.0010	64
1.00	8.99e-02	0.70	8	0.0005	256	1.00	5.81e-01	0.72	2	0.0010	64	0.99	0.00e+00	0.45	8	0.0001	256	0.56	8.28e-02	0.53	2	0.0000	32
1.00	0.00e+00	0.50	8	0.0001	256	0.77	7.94e-02	0.49	2	0.0000	32	1.00	3.18e-01	0.77	8	0.0050	256	1.00	2.82e-01	0.86	2	0.0005	32
0.99	3.67e-01	0.71	8	0.0050	256	1.00	4.70e-01	0.81	2	0.0005	32	1.00	5.25e-01	0.76	8	0.0010	256	0.97	3.21e-02	0.75	2	0.0001	32
1.00	4.21e-01	0.73	8	0.0010	256	0.98	2.36e-02	0.73	2	0.0001	32	0.81	9.71e-03	0.47	8	0.0000	128	1.00	6.03e-01	0.87	2	0.0050	32
0.93	2.53e-03	0.55	8	0.0000	128	1.00	7.84e-01	0.85	2	0.0050	32	1.00	5.19e-01	0.87	8	0.0005	128	1.00	5.15e-01	0.87	2	0.0010	32
1.00	3.94e-01	0.83	8	0.0005	128	1.00	7.15e-01	0.82	2	0.0010	32	0.99	0.00e+00	0.64	8	0.0001	128						
1.00	0.00e+00	0.74	8	0.0001	128	0.66	0.00e+00	0.42	32	0.0000	512	1.00	6.68e-01	0.89	8	0.0050	128						

E ABLATION OF RFDIFFUSION BLOCKS

We systematically ablate blocks in the RFdiffusion to localize the semantic encoding of our target properties. We measure the degradation of the generative capability by comparing the helix score (Figure 10) and SASA distribution statistics (Figure 11) of ablated models against the original. In both cases, block *main_04* is identified as the most critical component; its removal causes the most significant divergence from the baseline behavior, maximizing the score difference defined in Eq. **??**.

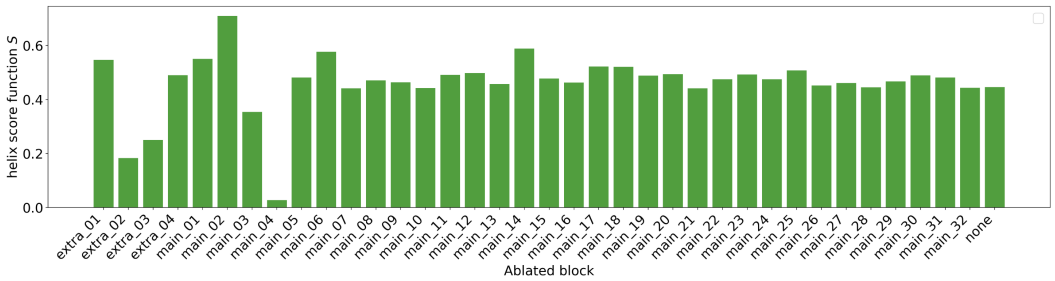


Figure 10: Localization of secondary structure encoding. Impact of systematic block ablation on secondary structure generation. The bar plot displays the helix score (fraction of residues in helices) for each configuration. Ablation of block *main_04* causes a collapse in helix generation, identifying it as the critical component for this property.

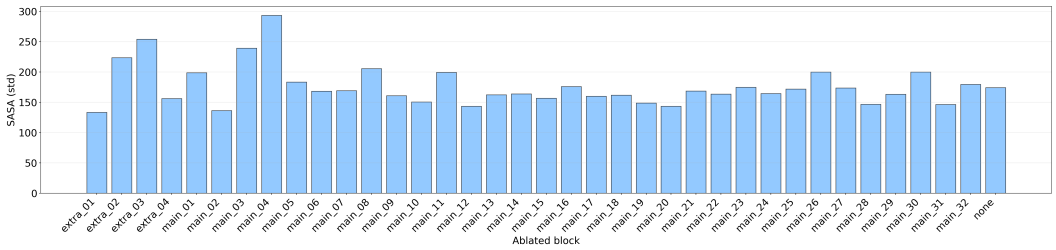


Figure 11: Localization of SASA encoding. Impact of systematic block ablation on Solvent Accessible Surface Area (SASA). The bar plot displays the SASA score function (standard deviation) for each configuration. Ablating block *main_04* results in the highest variance, indicating a loss of control over surface properties.

F FEATURE SELECTION

Dataset creation We gather a new dataset for the training of probing models and the analysis of their coefficients. First, using RFdiffusion with integrated SAE we generate 10000 of proteins without making any interventions and store the SAE encoder's activations together with their associated timesteps, proteins and residues. Using Stride, to each residue we assign secondary structure, and map these assignments to corresponding SAE activations. SASA annotations are on residue level using Mitternacht (2016). We visualize this approach in Appendix, (Figure 12). [PS sth wrong here]

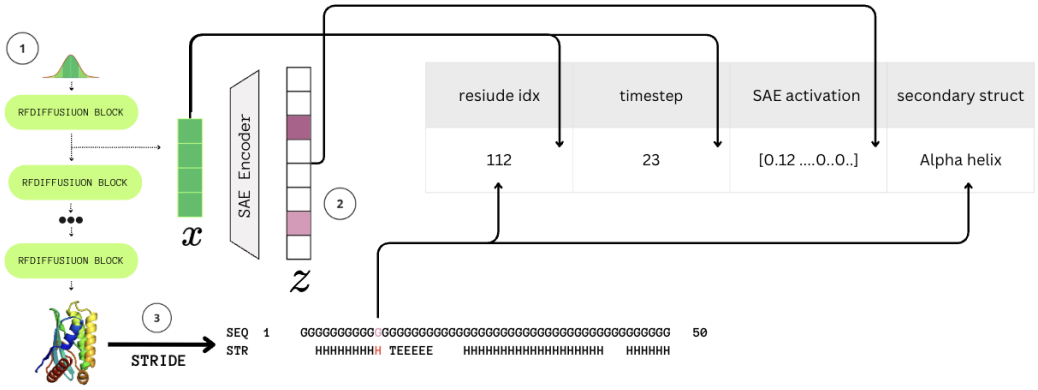


Figure 12: Dataset creation. 1) We generate proteins and cache activations of chosen block from each timestep, spliting them into patches per residue. 2) We reconstruct block activations with SAE and cache SAE encoder activations for each residue. 3) We assign secondary structure for each residue

Probing model training We develop binary classifiers in a One-vs-Rest (OvR) framework. For secondary structures, we train separate 'helix vs. rest' and 'strand vs. rest' models. To mitigate the class imbalance present in secondary structures (Figure 7), we apply class weighting during optimization.

In case of SASA we address it's continuous nature by discretizing the target by training classifiers to distinguish extreme quartiles: 'above 3$^{\text{rd}}$ quartile ($> Q3$) vs. rest' and 'below 1$^{\text{st}}$ quartile ($< Q1$) vs. rest'. We evaluate these classifiers in two configurations: *time-dependent* (trained on activations from specific timesteps) and *time-agnostic* (trained on activations pooled across all steps).

Notably, the models maintain robust performance even in the *time-agnostic* setting. We achieve ROC AUC scores of 94.1% and 93.3% for the helix and strand tasks, respectively. The SASA probes also perform reliably, reaching 81.78% AUC for the high-SASA ($> Q3$) detector and 78.86% for the low-SASA ($< Q1$) detector. In this section, we provide the detailed breakdown of classifier

performance across the diffusion trajectory, complementing the summary statistics provided in the main text. Figure 13 illustrates the stability of the probing models by comparing the *time-dependent* performance at each diffusion step ($t = 50 \rightarrow 1$) against the *time-agnostic* baseline.

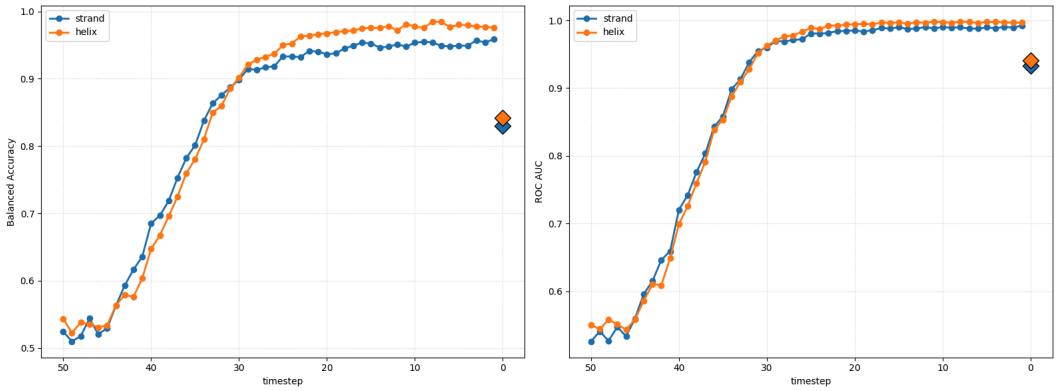


Figure 13: Results of training *time-agnostic* and *time-dependent* probing models. The first diffusion step is 50 and the last is 1; the diamond at 0 denotes the score for the *time-agnostic* model. The left pane reports balanced accuracy and the right pane reports AUC ROC. We observe robust performance of *time-agnostic models* compared to individual timesteps.

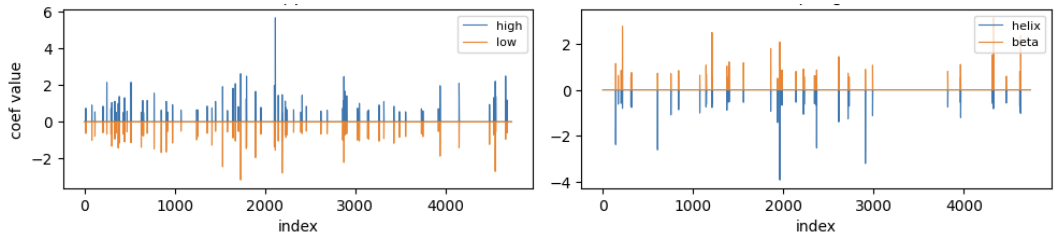


Figure 14: FoldSAE interpretation. Visualization of regression coefficients for the high (above 3rd quartile) classifiers - red and low (below 1str quartile) classifier - blue. Coefficients with an absolute magnitude greater than or equal to 0.1 are highlighted as solid lines, while those with magnitudes less than 0.1 are depicted as faint dashed lines. The largest coefficients often coincide at the same feature indices but exhibit opposite signs, suggesting a shared set of latent features governs the structural differentiation.

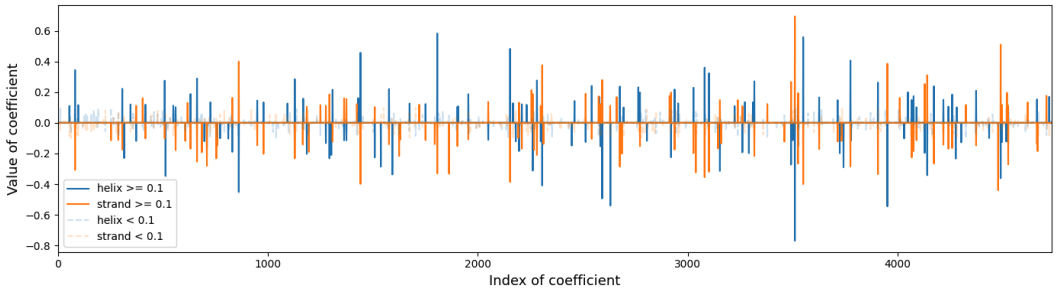


Figure 15: FoldSAE interpretation. Visualization of regression coefficients for the helix vs. rest (blue) and strand vs. rest (orange) probing classifiers. Coefficients with an absolute magnitude greater than or equal to 0.1 are highlighted as solid lines, while those with magnitudes less than 0.1 are depicted as faint dashed lines. The largest coefficients often coincide at the same feature indices but exhibit opposite signs, suggesting a shared set of latent features governs the structural differentiation.

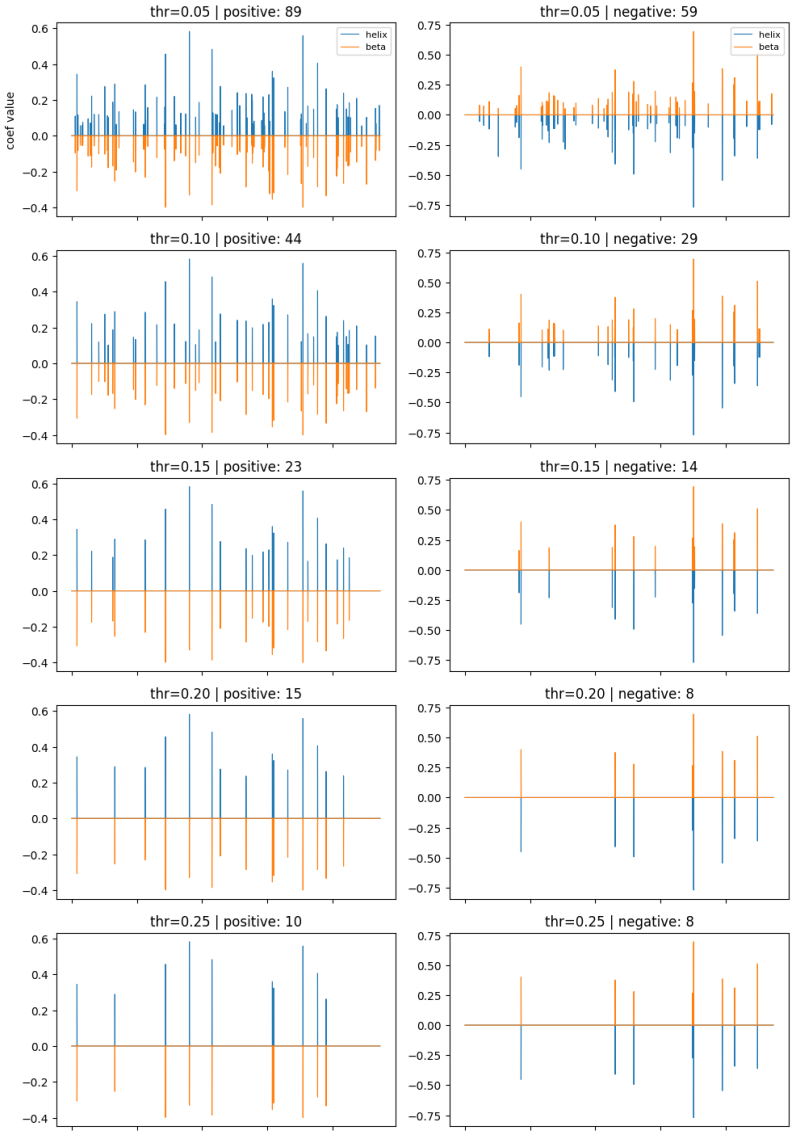


Figure 16: Visualization of the feature selection process using probing model coefficients. The panels illustrate the sparsification of features as the selection threshold increases (rows, from 0.05 to 0.35). Blue and orange bars represent coefficients for *alpha helix* and *beta sheet* classes, respectively. The subplot titles display the count of remaining discriminative features that satisfy two conditions: the coefficient modulus exceeds the chosen threshold, and the coefficients for the two classes exhibit opposite signs.

Features selection based on probing models One of aspect to consider after training probing models is threshold to choose most discriminative features. We pick only these features for which absolute value of corresponding feature is bigger than the threshold. Visualisation of number of features for each threshold can seen in Figure 16.