# ITERATIVE DISTILLATION OF MATRYOSHKA SPARSE AUTOENCODERS REVEALS FEATURE HIERARCHIES IN ESMFOLD2

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Matryoshka sparse autoencoders (SAEs) can decompose model representations into interpretable features, but reconstruction quality alone does not indicate how strongly those features influence model behavior. Furthermore, learned features can also vary across runs, complicating their characterization and reproducibility. These challenges are particularly difficult to study using large language models (LLMs), since SAE dictionary sizes capture only a fraction of an LLM's feature space. In contrast, protein folding models may provide a better test case because protein structure is organized across multiple scales, from individual amino acids and secondary structure to domains and global structure. Here, we apply Distilled Matryoshka SAEs (DMSAEs) to study activations from ESMC-6B in ESMFold2. We assign each latent an attribution score using gradient $\times$ activation with respect to ESMFold2's predicted local distance difference test (pLDDT) and predicted template modeling (pTM) scores. These metrics reflect the model's confidence in the local and global structure, respectively. We find that distillation improves the recovery and hierarchical placement of features associated with individual amino acids across independent training runs. We also find that DMSAE distillation produces more reproducible and clearly separated rankings of amino acid features than existing SAE baselines. These results show that Distilled Matryoshka SAEs can refine the coarse ordering imposed by nested Matryoshka prefixes and improve the consistency of learned feature hierarchies.

## 1 INTRODUCTION

Sparse autoencoders (SAEs) are increasingly used to decompose neural network activations into sparse, interpretable features. Beyond recovering individual features, an important goal is to organize them into hierarchies that reflect relationships across different levels of abstraction. Matryoshka sparse autoencoders (MSAEs) achieve this goal by training nested dictionary prefixes to reconstruct the same activation (Bussmann et al., 2025). This provides a coarse organization across dictionary scales, but does not impose fine-grained feature ordering or ensure consistent feature placement across runs, which has motivated alternative approaches such as Distilled Matryoshka SAEs (DMSAEs) (Martin-Linares & Ling, 2026), Ordered SAEs (Wang et al., 2025), and MP-SAEs (Costa et al., 2025).

Building on the DMSAE method (Martin-Linares & Ling, 2026), we adapt this framework into an iterative training procedure. Stable encoder directions are promoted into a frozen core during training, while the remaining SAE dictionary continues to learn the residual.

SAEs have been used extensively to study LLM representations, but the large feature space of LLMs and the difficulty of constructing reliable benchmarks for complex language features make systematic evaluation challenging. Protein folding models may be easier to evaluate because proteins have well-defined features across several biological scales, from individual amino acids and secondary structure to domains and global structure. These established feature groupings offer a more constrained setting for studying SAE hierarchies than the open-ended feature space of LLMs (Karvonen et al., 2025; Chanin, 2026). Previous studies have applied SAEs to study features associated with individual residues and annotated protein domains (Simon & Zou, 2025; Parsan et al., 2025; Liu et al.,



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

2026). Here, we apply DMSAE to ESMC-6B activations used by ESMFold2 and use individual amino acids as a broad feature group that represents the most basic level of protein representation.

While examining a publicly released Matryoshka SAE trained on ESM2-3B representations used by ESMFold (Parsan et al., 2025), we noticed that closely related amino acid features could be assigned to different levels of the learned hierarchy. Using the interactive web application Parsan et al. (2025), we identified isoleucine and leucine-specific latents in different Matryoshka groups (Fig. 1). Leucine and isoleucine are chemically similar, yet their corresponding features appear in different Matryoshka prefixes. This motivated us to ask whether this difference reflects an important distinction in the underlying model representation, or an artifact of the coarse hierarchy imposed by nested Matryoshka prefixes.

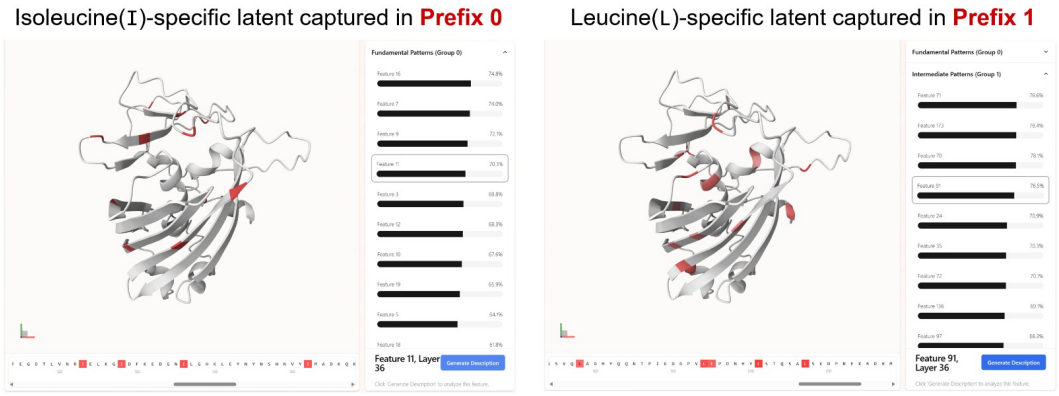


Figure 1: **Similar amino acid features can appear at different levels of the Matryoshka SAE hierarchy.** In a Matryoshka SAE trained on ESM2-3B representations used by ESMFold (Parsan et al., 2025), an isoleucine-specific latent appears in Prefix 0, while a leucine-specific latent appears in Prefix 1. Leucine and isoleucine have similar chemical properties, so their separation across prefixes may reflect an important difference in the underlying model representation or variability introduced by the coarse structure of nested Matryoshka prefixes. Visualizations were obtained from the interactive web application released with Parsan et al. (2025).

To test whether progressive distillation improves hierarchical organization, we trained DMSAEs on ESMC-6B activations used by ESMFold2 and compared them with vanilla MSAEs across independent runs. DMSAE training improved amino acid feature recovery and placed these features more consistently in the retained core. We observed similar improvements with reconstruction-guided and random promotion. This result shows that improved hierarchical placement does not depend on task-specific pLDDT or pTM attribution. Comparisons with other SAE methods also showed that feature recovery alone does not capture representation quality. A feature can be recovered in the dictionary but still be split across several latents. Together, these results show that progressive distillation can improve the consistency and compactness of hierarchical SAE representations.

## 2 Related Work

**Evaluating sparse autoencoders.** Most SAE development has focused on sparsity and reconstruction quality, but neither directly measures feature interpretability or usefulness. SAEBench introduced a broader set of SAE evaluations that includes reconstruction, disentanglement, probing, and behavioral tests (Karvonen et al., 2025). Its results show that better performance on common SAE metrics does not always lead to better practical performance. A later study found that several SAEBench metrics were unreliable or unable to clearly distinguish between methods (Chanin, 2026). Synthetic benchmarks avoid some of these problems because the true features are known, but they do not reflect representations learned by real language or protein models. We instead evaluate SAEs in a real protein model using amino acid features with known labels and reproducible reference directions.

**Protein-model SAEs.** InterPLM identified biological SAE features in ESM2 and uses them for annotation and intervention (Simon & Zou, 2025). Matryoshka SAEs have also been applied to ESM2-3B representations used by ESMFold, where they recover features related to residue identity and protein structure (Parsan et al., 2025). ProtSAE uses biological annotations and domain knowledge during training (Liu et al., 2026). In our work, amino acid labels are used only for evaluation and do not affect SAE training or feature promotion.

**Hierarchical and ordered SAEs.** Matryoshka SAEs train nested dictionary prefixes to reconstruct the same activation (Bussmann et al., 2025). Ordered SAEs encourage a finer ordering of features (Wang et al., 2025), while Hierarchical SAE learns explicit parent–child relationships between features (Luo et al., 2026). MP-SAE instead changes inference by repeatedly selecting features against the remaining residual (Costa et al., 2025). DMSAE keeps the standard sparse reconstruction objective but gradually freezes selected encoder directions during training.

**Feature consistency and downstream relevance.** Independent SAE runs can learn different features even when trained on the same model and data (Paulo & Belrose, 2026; Leask et al., 2025). Stable subspaces can also coexist with unstable individual features (Gerasimov et al., 2026). In addition, an interpretable feature does not necessarily have the same downstream effect in every context (Hoang et al., 2026). We therefore evaluate feature recovery and downstream attribution separately.

# 3 METHODS

## 3.1 MATRYOSHKA SPARSE RECONSTRUCTION

We train SAEs on raw ESMC-6B residue activations used by ESMFold2. For activation $x_i \in \mathbb{R}^d$, unmasked latent responses are

$$a_i = \mathrm{ReLU}(W_{\mathrm{enc}}(x_i - b_{\mathrm{dec}}) + b_{\mathrm{enc}})\,. \tag{1}$$

A BatchTopK mask is applied to the trainable non-core latents. The resulting sparse code is shared across all Matryoshka prefixes, and each prefix reconstructs the input using the retained core plus the corresponding prefix of the non-core dictionary. The primary objective is the mean reconstruction error over prefixes,

$$\mathcal{L}_{\mathrm{prefix}} = \frac{1}{LBd} \sum_{\ell=1}^{L} \sum_{i=1}^{B} \left\| x_i - \hat{x}_i^{(\ell)} \right\|_2^2, \tag{2}$$

with an AuxK term for inactive non-core features. Full masking, AuxK, and optimization details are provided in Appendix D.

## 3.2 PROGRESSIVE DISTILLATION

DMSAE divides the dictionary into a retained core and a trainable non-core. Every DMSAE is first trained for 10,000 reconstruction steps with batch size 4,096 before any promotion is allowed. Thereafter, only latents in the innermost Matryoshka group (Prefix 0) are eligible for promotion. Promotion is considered only after the Prefix 0 encoder directions satisfy the cosine-similarity stability gate, with at least 100 optimization steps between successive promotions.

When a promotion event occurs, one eligible Prefix 0 encoder direction is moved into the core and frozen. Its decoder column and bias remain trainable. This preserves the selected detector while allowing its reconstruction contribution to adapt as the surrounding representation changes. Because core latents contribute to every prefix, subsequent optimization trains the non-core against a changed residual reconstruction problem. After each promotion, a new latent is initialized in Prefix 0 and one non-core latent is removed, preserving the width of Prefix 0 and the total dictionary size. Further implementation details are provided in Appendix D.

## 3.3 PLDDT/PTM ATTRIBUTION

For calibration protein $p$, let $h_{p,r}$ be the activation of residue $r$ at the SAE site. We construct differentiable confidence scalars $F_q$ for $q \in \{\mathrm{pLDDT}, \mathrm{pTM}\}$ from ESMFold2 outputs and cache

$$g_{p,r}^{(q)} = \nabla_{h_{p,r}} F_q. \tag{3}$$

ESMC and ESMFold2 parameters remain frozen. For latent $j$, attribution is

$$A_j^{(q)} = \frac{1}{N_{\text{cal}}} \sum_p \sum_{r \in \mathcal{R}_p} \left| a_j (h_{p,r}) (g_{p,r}^{(q)})^\top d_j \right|, \tag{4}$$

where $d_j$ is the decoder direction. This is a local gradient-times-activation sensitivity magnitude, not a finite intervention effect.

For the primary method, stable candidates are ranked by

$$P_j = S_j \sqrt{\frac{A_j^{(\text{pLDDT})}}{\max_k A_k^{(\text{pLDDT})}} \frac{A_j^{(\text{pTM})}}{\max_k A_k^{(\text{pTM})}}}, \tag{5}$$

where $S_j$ measures recent directional stability. The highest-scoring eligible candidate is promoted.

### 3.4 AMINO ACID REFERENCE DIRECTIONS

For each of the 20 canonical amino acids, we train 100 independent ReLU probes from the same raw ESMC activation coordinates used by the SAE. Let $v_{a,r}$ be the normalized encoder direction for amino acid $a$ in probe run $r$. We define the consensus direction

$$p_a = \frac{\sum_{r=1}^{R} v_{a,r} / \|v_{a,r}\|_2}{\left\| \sum_{r=1}^{R} v_{a,r} / \|v_{a,r}\|_2 \right\|_2}. \tag{6}$$

For SAE encoder row $w_j$, directional agreement is

$$s_{a,j} = p_a^\top \frac{w_j}{\|w_j\|_2}. \tag{7}$$

The probes are used only for evaluation; probe labels and directions never enter SAE training or promotion.

### 3.5 BASELINES AND EVALUATION

We benchmark six SAE configurations using 10 random seeds per configuration. **Vanilla MSAE** serves as the baseline and is trained with the standard Matryoshka reconstruction objective, without moving selected latents into the frozen core. **pLDDT/pTM DMSAE** uses Eq. equation 5 to select among eligible Prefix 0 candidates. **Reconstruction DMSAE** promotes the eligible Prefix 0 latent whose ablation would cause the largest increase in reconstruction error. **Random DMSAE** selects uniformly at random from eligible Prefix 0 candidates rather than ranking them by an attribution score. **Ordered SAE** retains the $k$ most active latents for each input and trains nested dictionary prefixes, increasing in size by one latent, to reconstruct that input (Wang et al., 2025). **MP-SAE** builds the reconstruction one latent at a time, selecting each additional latent to explain the part of the input not yet reconstructed (Costa et al., 2025). Additional baseline details are provided in Appendix E.

For each amino acid and seed-level run, we record: (i) the maximum cosine similarity to any SAE latent; (ii) the hierarchical location of the best-matching latent; (iii) the number of latents with cosine similarity greater than $0.5$; and (iv) the pLDDT/pTM attribution rank of amino acid-aligned latents (the last value ranks all dictionary latents by downstream attribution). For Figures 6 and 7, all pLDDT/pTM attributions are recomputed post hoc from the final checkpoint of every model, including the pLDDT/pTM-guided DMSAE.

We trained all models on NVIDIA RTX PRO 6000 GPUs for 50,000 optimization steps, corresponding to 205 million activations seen during training. We used the same training budget for all methods to support direct comparison. For DMSAE models, promotion was disabled for the first 10,000 steps and allowed from step 10,000 through step 50,000.

# 4 RESULTS

## 4.1 AMINO ACID PROBES PROVIDE REPRODUCIBLE REFERENCE DIRECTIONS

We first tested whether amino acid identity yields a stable reference direction in ESMC activation space. Across 100 independently trained probes per amino acid, encoder directions align strongly with their amino acid consensus directions (Fig. 2). Most cosine similarities lie between approximately 0.8 and 0.9, although glutamine shows greater variability. Probe predictions are also highly correlated across independent runs, with Pearson correlations ranging from 0.974 to 0.997 across amino acids (Appendix A). We used these vectors as reference directions for SAE evaluation.

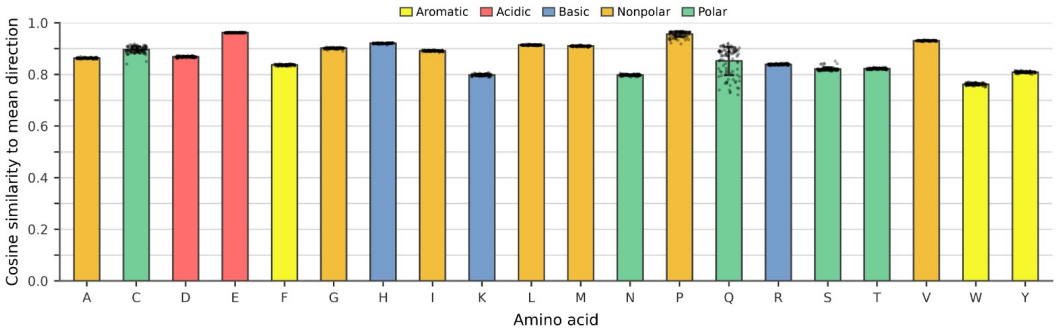


Figure 2: **Independent amino acid probe directions provide reference directions for evaluating whether SAE latents recover amino acid-specific representations.** For each amino acid, we trained 100 independent probes and defined a consensus direction from their normalized encoder vectors. Individual probe directions generally aligned closely with these consensus directions, although they do not converge to a single identical direction since cosine similarities range from 0.8–0.9. This variation suggests that multiple encoder directions can achieve similarly high predictive performance, consistent with the high Pearson correlations reported in Table 1. Interestingly, glutamine (Q) showed a substantially wider distribution of cosine similarities than the other amino acids, for reasons that are unclear. Bars indicate mean, error bars show $\pm 1$ SD, and colors indicate amino acid chemistry. We used these amino acid consensus directions to evaluate SAE feature recovery in later analyses.

## 4.2 LATENT PROMOTION TO THE CORE LEADS TO REORGANIZATION OF AMINO ACID FEATURES INTO THE MOST NESTED HIERARCHY

We next located, for each amino acid and each seed, the SAE latent with maximum cosine similarity to the corresponding reference direction. In vanilla MSAEs, best-matching amino acid latents occur across several reconstruction prefixes (Fig. 3). Thus, even for a concept as general as amino acid identity, the nested reconstruction objective does not produce a stable hierarchical location across independent runs.

Progressive distillation substantially changes this pattern. Most amino acid reference directions are represented in the retained core under all three progressive policies, including pLDDT/pTM-guided, reconstruction-guided, and random promotion. This is important because only Prefix 0 latents are ever eligible for promotion. An amino acid direction represented only in an outer prefix of a vanilla MSAE cannot simply be transferred directly into the core. For that direction to appear in the DM-SAE core, continued training must first produce a matching Prefix 0 latent that can later be promoted. Figure 3 is therefore consistent with the frozen core changing the residual reconstruction problem and allowing amino acid features to be reallocated toward Prefix 0 as training proceeds. Direct longitudinal tracking of individual feature directions across promotion events would be needed to establish this migration process more explicitly. All three promotion policies improved hierarchical placement, including random promotion. This suggests that the improvement comes mainly from progressively freezing stable features in the core rather than from the specific score used to select them.




Figure 3: **Distillation restores hierarchical consistency for amino acid features that vanilla Matryoshka SAEs place inconsistently across prefixes.** For each amino acid, we identified the best-matching latent across all latents and recorded its location in the Matryoshka hierarchy across runs. In the vanilla Matryoshka SAE, amino acid-specific latents appeared across multiple prefixes, indicating hierarchical inconsistency even when the corresponding feature was learned. We then used iterative DMSAE distillation, which transfers selected encoder directions into a shared core across successive training cycles. Distillation concentrated most amino acid features in the core whether promotion from prefix 0 was guided by pLDDT/pTM attribution, reconstruction attribution, or random selection. Because amino acid identity represents the most minimal element of a protein sequence, these results suggest that iterative distillation can recover general features in the earliest level of the hierarchy, even without task-specific attribution.

### 4.3 MP-SAE MAXIMIZES BEST-LATENT ALIGNMENT BUT FRAGMENTS AMINO ACID DIRECTIONS

We then compared how accurately each architecture recovers the amino acid reference directions, independent of hierarchical location. Most Matryoshka-based methods obtain similar maximum cosine similarity (Fig. 4). The DMSAE variants are slightly higher on average than vanilla MSAE, although these differences are not statistically significant in the current analysis. Ordered SAE is lower, whereas MP-SAE achieves the highest maximum cosine similarity and approaches the empirical probe-to-consensus ceiling from Fig. 2.

Maximum similarity alone, however, hides how the representation is distributed across the dictionary. Figure 5 counts the number of latents with cosine similarity at least $0.5$ to each amino acid reference direction. Vanilla MSAE and Ordered SAE fail to recover a matching latent for roughly $30\%$ of amino acid observations across seeds. The DMSAEs recover a match for approximately $90\%$ of observations. MP-SAE also recovers most amino acid directions, but frequently with several matching latents (in some cases, more than 20 latents align to one amino acid direction). Thus,

MP-SAE is highly effective at recovering a direction somewhere in the dictionary, but the same biological concept can be distributed redundantly across many latents.

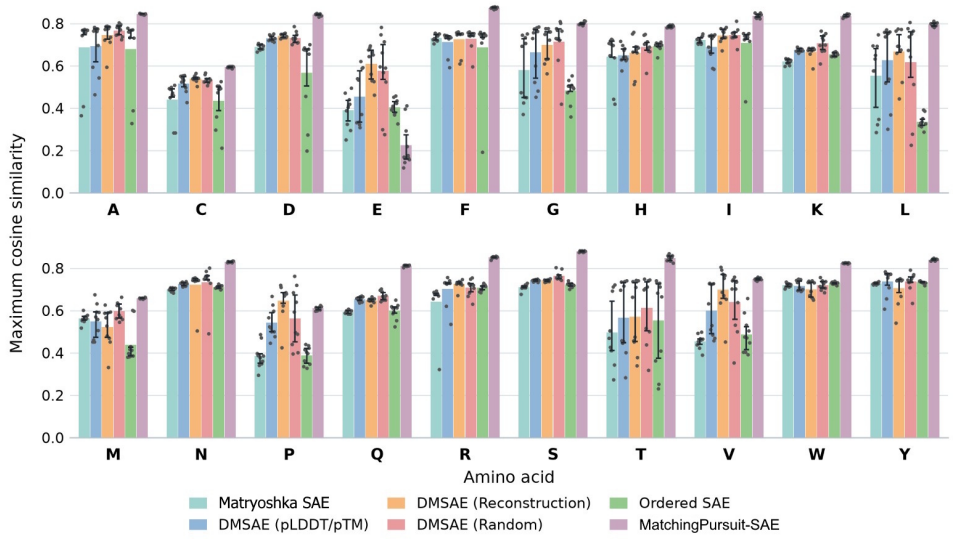


Figure 4: **Most Matryoshka-based SAEs recover amino acid directions with similar cosine similarity.** The DMSAE variants slightly outperform vanilla Matryoshka SAE on average, although the differences are not statistically significant. Ordered SAE shows lower alignment, whereas MP-SAE achieves substantially higher cosine similarity, approaching the empirical ceiling suggested by Fig. 2.

### 4.4 Post hoc attribution becomes more separable after progressive distillation

We next asked how the recovered features relate to ESMFold2's confidence calculations. For every latent with cosine similarity greater than $0.5$ to an amino acid consensus direction, we recomputed joint pLDDT/pTM attribution from the final checkpoint and ranked the latent against the complete SAE dictionary. This post hoc calculation was performed identically for every architecture so the values shown here are not the scores used during training for latent promotion.

Vanilla MSAE shows substantial variation and overlap in attribution rank across seeds and amino acids (Fig. 6). Ordered SAE also shows broad variability. In contrast, pLDDT/pTM-guided and reconstruction-guided DMSAEs produce tighter distributions and more separable amino acid-associated ranks. Random-promotion DMSAE is less structured, with many amino acid features occupying similar rank ranges, while MP-SAE exhibits the widest variation, consistent with the redundant amino acid representation seen in Fig. 5. The more consistent attribution ranks show that DMSAE models place amino acid features more reproducibly across runs. However, they do not show that the resulting order reflects the model's underlying ground truth ranking, which remains a limitation of our current methods.

Nevertheless, these results show why attribution should be considered together with feature recovery and feature splitting. In MP-SAE, low per-latent attribution can result from the same feature being spread across several latents rather than from weak overall influence (Figs. 5 and 7). Ordered SAE shows a different issue because its attribution scores are harder to interpret when the corresponding latents align less closely with the amino acid reference directions.

Our experiments test only amino acid identity, where labels and reference directions are well defined. Future work should test physicochemical residue classes, secondary structure, domains, and global folds. This would show whether SAE feature ordering follows increasing levels of biological organization.

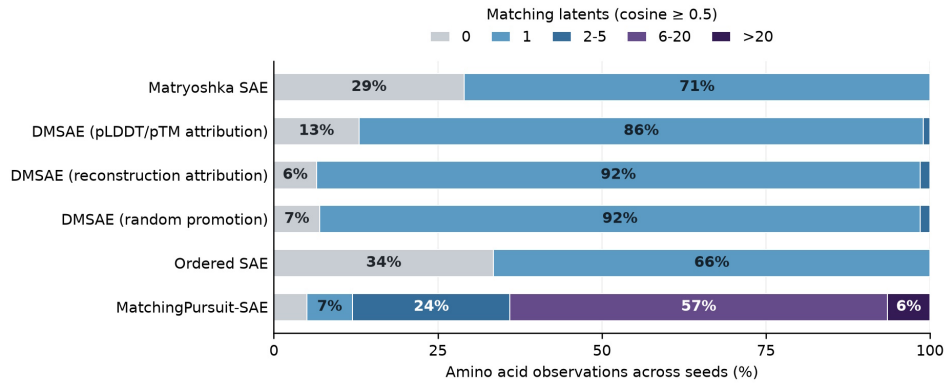


Figure 5: **DMSAEs improve amino acid feature recovery, while MP-SAE shows substantial feature splitting.** Vanilla Matryoshka and Ordered SAEs fail to recover a latent with cosine similarity $\geq 0.5$ for $\sim$30% of amino acid observations across seeds. In contrast, all DMSAE variants capture $\sim$90% of amino acids at cosine similarity $\geq 0.5$. MP-SAE also recovers most amino acid directions and achieves high maximum cosine similarity, but often represents the same direction with many latents (in some cases, more than 20 latents align with a single amino acid consensus direction). This feature splitting makes it harder to identify a unique relevant latent in an unsupervised setting.

## 5 Limitations

Our evaluation uses amino acid consensus directions learned from independent probes. These directions are reproducible but not unique, so they provide useful references rather than ground truth. We also evaluate only amino acid identity, which represents the most basic level of protein organization. We therefore do not establish whether DMSAE recovers consistent hierarchies across higher-level features such as secondary structure, domains, or global folds.

Our gradient-times-activation score measures local sensitivity to ESMFold2 pLDDT and pTM outputs. It does not measure causal effects or provide a known ground-truth ordering of amino acid importance (Hoang et al., 2026). In addition, the compared SAE methods use different inference procedures and can have different realized activity at the same nominal sparsity. Comparisons at precisely equivalent $L_0$ would provide a more controlled test across architectures, however matching activity levels across methods is difficult in practice.

## 6 Conclusion

Protein models provide well-defined biological features that can be used to evaluate SAE hierarchies. Using amino acid identity as a labeled reference, we find that vanilla Matryoshka SAEs place these features inconsistently across prefixes. DMSAE training improves both feature recovery and hierarchical placement across pLDDT/pTM-guided, reconstruction-guided, and random promotion. These results suggest that progressively freezing stable features in the core drives much of this improvement. Our comparisons with MP-SAE and Ordered SAE also show that feature recovery should be evaluated together with feature splitting and downstream attribution. Future work can extend this analysis to secondary structure, domains, and global folds to test whether SAE hierarchies also recover higher levels of protein organization.

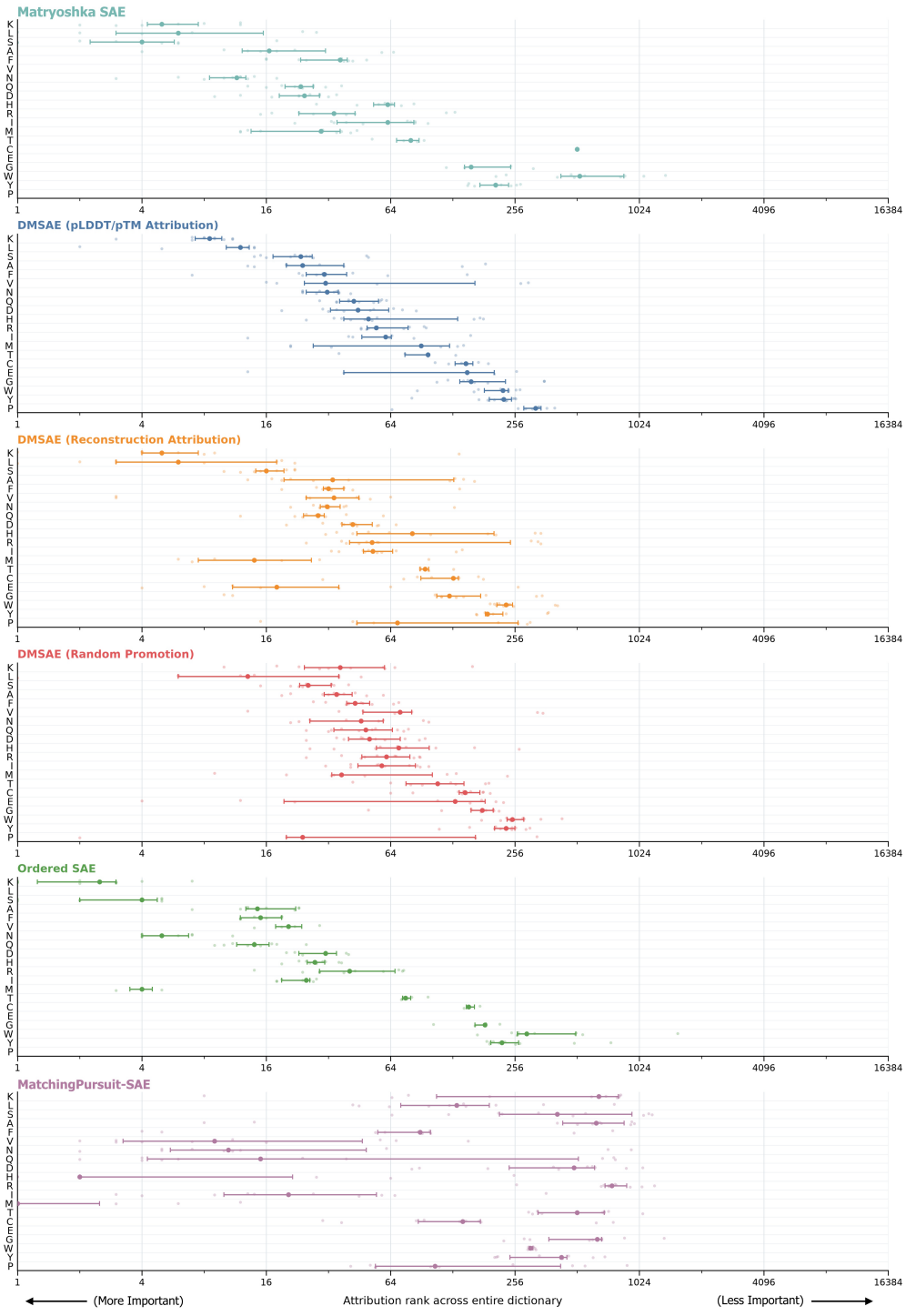


Figure 6: **DMSAE training produces a more consistent ordering of amino acid features by attribution rank.** For each amino acid, we show the attribution ranks of latents with cosine similarity $> 0.5$ to its consensus direction, with ranks computed over the full SAE dictionary on a log scale. The pLDDT/pTM- and reconstruction-guided DMSAEs show lower variability across seeds and a clearer progression in attribution rank across amino acids than the vanilla Matryoshka and Ordered SAEs. This ordering is not perfect, indicating that the learned core does not fully eliminate variability. Random-promotion DMSAE shows a weaker ordering, while MP-SAE shows substantially greater variation in attribution rank, consistent with its more redundant representation of amino acid directions.

## 7 AI ASSISTANCE

Generative AI tools were used to polish writing and assist with literature retrieval and discovery. All AI-assisted content was reviewed by the authors, who take responsibility for the final manuscript.

## REFERENCES

Bart Bussmann, Noa Nabeshima, Adam Karvonen, and Neel Nanda. Learning multi-level features with Matryoshka sparse autoencoders. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 6077–6101. PMLR, 2025. URL https://proceedings.mlr.press/v267/bussmann25a.html.

David Chanin. Are sparse autoencoder benchmarks reliable? *arXiv preprint arXiv:2605.18229*, 2026. URL https://arxiv.org/abs/2605.18229.

Valérie Costa, Thomas Fel, Ekdeep Singh Lubana, Bahareh Tolooshams, and Demba Ba. From flat to hierarchical: Extracting sparse representations with matching pursuit. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-1154. URL https://openreview.net/forum?id=Ll5miDx8KB.

Gleb Gerasimov, Timofei Rusalev, Nikita Balagansky, Daniil Laptev, Vadim Kurochkin, and Daniil Gavrilov. Unstable features, reproducible subspaces: Understanding seed dependence in sparse autoencoders, 2026. URL https://arxiv.org/abs/2606.12138v1. Version 1.

Phu Gia Hoang, Anwoy Chatterjee, Tanmoy Chakraborty, Iryna Gurevych, and Subhabrata Dutta. Sparse autoencoders encode both concepts and functions: The downstream geometry of feature effects, 2026. URL https://arxiv.org/abs/2607.24645v1. Version 1.

Adam Karvonen, Can Rager, Johnny Lin, Curt Tigges, Joseph Bloom, David Chanin, Yeu-Tong Lau, Eoin Farrell, Callum McDougall, Kola Ayonrinde, Demian Till, Matthew Wearden, Arthur Conmy, Samuel Marks, and Neel Nanda. SAEBench: A comprehensive benchmark for sparse autoencoders in language model interpretability. In *Proceedings of the 42nd International Conference on Machine Learning*, 2025. URL https://arxiv.org/abs/2503.09532.

Patrick Leask, Bart Bussmann, Michael Pearce, Joseph Bloom, Curt Tigges, Noura Al Moubayed, Lee Sharkey, and Neel Nanda. Sparse autoencoders do not find canonical units of analysis. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2502.04878.

Xiangyu Liu, Haodi Lei, Yi Liu, Yang Liu, and Wei Hu. ProtSAE: Disentangling and interpreting protein language models via semantically-guided sparse autoencoders, 2026. URL https://arxiv.org/abs/2509.05309v2. Version 2.

Yifan Luo, Yang Zhan, Jiedong Jiang, Tianyang Liu, Mingrui Wu, Zhennan Zhou, and Bin Dong. From atoms to trees: Building a structured feature forest with hierarchical sparse autoencoders, 2026. URL https://arxiv.org/abs/2602.11881.

Cristina P. Martin-Linares and Jonathan P. Ling. Iterative distillation for feature-consistent sparse autoencoders. In *Mechanistic Interpretability Workshop at the 43rd International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=NWEQ9wqhfM.

Nithin Parsan, David J. Yang, and John J. Yang. Towards interpretable protein structure prediction with sparse autoencoders. In *GEM Workshop at ICLR*, 2025. URL https://arxiv.org/abs/2503.08764v1.

Gonçalo Paulo and Nora Belrose. Sparse autoencoders trained on the same data learn different features. In *International Conference on Learning Representations*, 2026.

Elana Simon and James Zou. InterPLM: Discovering interpretable features in protein language models via sparse autoencoders. *Nature Methods*, 22:2107–2117, 2025. doi: 10.1038/s41592-025-02836-7.

Sophie L. Wang, Alex Quach, Nithin Parsan, and John J. Yang. Enforcing orderedness in SAEs to improve feature consistency, 2025. URL https://arxiv.org/abs/2512.02194.

## A PROBE PREDICTION CONSISTENCY

Table 1: Pearson correlation coefficients for independently trained amino acid probes.

| **Amino acid** | **A** | **C** | **D** | **E** | **F** | **G** | **H** | **I** | **K** | **L** |
|---|---|---|---|---|---|---|---|---|---|---|
| **Pearson** $r$ | 0.993 | 0.974 | 0.996 | 0.992 | 0.990 | 0.992 | 0.989 | 0.994 | 0.997 | 0.994 |

| **Amino acid** | **M** | **N** | **P** | **Q** | **R** | **S** | **T** | **V** | **W** | **Y** |
|---|---|---|---|---|---|---|---|---|---|---|
| **Pearson** $r$ | 0.988 | 0.996 | 0.985 | 0.987 | 0.993 | 0.997 | 0.996 | 0.993 | 0.996 | 0.994 |

## B RAW DOWNSTREAM ATTRIBUTION MAGNITUDES

Figure 7 reports the attribution-score magnitudes corresponding to the rank analysis in Fig. 6. As in Figure 6, all scores are recomputed post hoc from the final checkpoint for every architecture; the pLDDT/pTM DMSAE does not reuse the scores that triggered promotion during training. Rank is emphasized in the main text because raw gradient-times-activation scale can differ across SAE architectures and training states, whereas within-dictionary rank directly asks where an amino acid-associated latent lies relative to the other latents in the same model. Importantly, neither raw score nor rank can be interpreted independently of feature recovery and feature splitting.

## C DMSAE TRAINING PSEUDOCODE

**Algorithm 1** DMSAE training

**Require:** Activations $\mathcal{D}$, prefix widths $\{m_\ell\}_{\ell=1}^{L}$, sparsity $k$, maximum core size $C_{\max}$

Initialize DMSAE parameters and $c \leftarrow 0$
**for** $t = 1, 2, \ldots$ **do**
  Sample $X \sim \mathcal{D}$
  $Z \leftarrow \mathrm{ReLU}\big((X - \mathbf{b}_{\mathrm{dec}})W_{\mathrm{enc}}^{\top} + \mathbf{b}_{\mathrm{enc}}\big)$
  $Z_{\mathrm{core}} \leftarrow Z_{[:,1:c]}, \quad \widetilde{Z}_{\mathrm{nc}} \leftarrow \mathrm{BatchTopK}\big(Z_{[:,c+1:]}, k\big)$
  $\mathcal{L}_{\mathrm{rec}} \leftarrow \frac{1}{L}\sum_{\ell=1}^{L}\Big\|X - \Big(\mathbf{b}_{\mathrm{dec}} + Z_{\mathrm{core}}W_{\mathrm{dec},[:,1:c]}^{\top} + \widetilde{Z}_{\mathrm{nc},[:,1:m_\ell]}W_{\mathrm{dec},[:,c+1:c+m_\ell]}^{\top}\Big)\Big\|_2^2$
  Optimize $\mathcal{L}_{\mathrm{rec}}$ with core encoder directions frozen
  Update stability scores $S_j$ of candidate encoder latents
  **if** candidate latents are stable and $c < C_{\max}$ **then**
    $j^\star \leftarrow$ SELECTPROMOTIONLATENT
    Promote $j^\star$; $c \leftarrow c + 1$
  **end if**
**end for**

**Algorithm 2** Latent Promotion (SELECTPROMOTIONLATENT)

**for** each candidate latent $j$ **do**
  Compute pLDDT and pTM attribution scores $A_j^{\mathrm{pLDDT}}$ and $A_j^{\mathrm{pTM}}$
**end for**
Normalize attribution scores across candidate latents
$j^\star \leftarrow \arg\max_j S_j \sqrt{\widetilde{A}_j^{\mathrm{pLDDT}}\,\widetilde{A}_j^{\mathrm{pTM}}}$
**return** $j^\star$

# D IMPLEMENTATION DETAILS

## D.1 CORE AND NON-CORE CODE

For core size $c$, let $\mathcal{C} = \{1, \dots, c\}$ and $\mathcal{N} = \{c+1, \dots, M\}$. The non-core sparsity budget is

$$k_{\mathrm{nc}}(c) = \left\lfloor k_{\mathrm{base}} \frac{M-c}{M} + \frac{1}{2} \right\rfloor. \tag{8}$$

For activation matrix $A \in \mathbb{R}^{B \times M}$,

$$Z_{\mathcal{N}} = \mathrm{BatchTopK}_{k_{\mathrm{nc}}(c)}(A_{\mathcal{N}}), \qquad Z_{\mathcal{C}} = A_{\mathcal{C}}. \tag{9}$$

Core coordinates therefore bypass non-core sparse competition.

Let $(g_1, \dots, g_L)$ be the initial non-core group widths. Core growth reduces only the final group. The cumulative non-core prefix widths are

$$m_\ell(c) = \begin{cases} \sum_{r=1}^{\ell} g_r, & \ell < L, \\ M - c, & \ell = L. \end{cases} \tag{10}$$

Every prefix uses the same sparse code,

$$\hat{x}_i^{(\ell)} = b_{\mathrm{dec}} + \sum_{j=1}^{c} a_{ij} d_j + \sum_{r=1}^{m_\ell(c)} z_{i,c+r} d_{c+r}. \tag{11}$$

## D.2 AUXK

Let $\mathcal{D}_t$ contain non-core latents that have not fired for at least $R_{\mathrm{dead}}$ activation rows. Auxiliary coefficients are selected per example from the largest unmasked activations restricted to $\mathcal{D}_t$. With

$$r_i = \mathrm{sg}(x_i - \hat{x}_i^{(L)}), \qquad \hat{r}_i^{\mathrm{aux}} = \sum_{j \in \mathcal{D}_t} v_{ij} d_j, \tag{12}$$

we use

$$\mathcal{L}_{\mathrm{AuxK}} = \frac{B^{-1} \sum_i \|r_i - \hat{r}_i^{\mathrm{aux}}\|_2^2}{B^{-1} \sum_i \|r_i - \bar{r}\|_2^2}, \qquad \bar{r} = B^{-1} \sum_i r_i. \tag{13}$$

The total objective is $\mathcal{L} = \mathcal{L}_{\mathrm{prefix}} + \alpha \mathcal{L}_{\mathrm{AuxK}}$. No confidence-attribution term is backpropagated through this objective.

## D.3 STABILITY GATE

At step $t$, the candidate pool is the current innermost non-core group $\mathcal{P}_t = \{c+1, \dots, c+g_1\}$. Let $u_{j,t} = w_{j,t} / \|w_{j,t}\|_2$. We track

$$S_j(t) = \min_{r=0,\dots,W-1} \langle u_{j,t-r}, u_{j,t-r-1} \rangle. \tag{14}$$

Attribution is attempted only when

$$\min_{j \in \mathcal{P}_t} S_j(t) \geq \tau_{\mathrm{stab}} \tag{15}$$

and warmup, cooldown, retry, and maximum-core-size conditions permit it.

## D.4 OPTIMIZATION AND HYPERPARAMETERS

Decoder columns are kept unit-normalized. Decoder gradients are projected onto the tangent space before the optimizer update and columns are renormalized afterward. Core encoder gradients are zeroed and saved encoder rows are restored exactly after each update. Per-latent Adam state follows latent identities during relocation.

For each new latent, the encoder and decoder vectors are initialized from the same sample $\xi \sim \mathcal{N}(0, I_d)$, normalized to unit length. The encoder bias is set to zero, and the optimizer state and activity counters are reset.

For the reported experiments, $d = 2560$, $M = 16384$, and the initial group widths are $(512, 1024, 2048, 4096, 8704)$. We use $k_{\text{base}} = 10$ and a maximum core size of 512. The stability window contains three comparisons with threshold $\tau_{\text{stab}} = 0.995$. Promotion is disabled before step 10,000. AuxK uses weight $1/32$, dead-feature threshold 2,048,000 activation rows, and $k_{\text{aux}} = 1280$. Training uses batches of 4,096 activation rows, learning rate $3 \times 10^{-4}$, and gradient clipping at 1.0. Promotion cooldown and failed-attribution retry intervals are both 100 steps.

## E BASELINE DETAILS

**Warm start.** Vanilla MSAE, reconstruction-guided DMSAE, and random-promotion DMSAE are initialized from the seed-matched 10,000-step parameter checkpoint of the primary training trajectory. The warm start copies $W_{\text{enc}}$, $W_{\text{dec}}$, $b_{\text{enc}}$, and $b_{\text{dec}}$ but resets the core layout, latent identities, firing counters, optimizer state, and progressive-training state.

**Reconstruction-guided DMSAE.** For innermost candidate $j$, let $r_i^{(\ell)} = x_i - \hat{x}_i^{(\ell)}$ and $\bar{r}_i = L^{-1} \sum_\ell r_i^{(\ell)}$. The fixed-code ablation increase is

$$\Delta_j^{\text{rec}} = \frac{1}{Bd} \sum_i \left( 2 z_{ij} \bar{r}_i^\top d_j + z_{ij}^2 \|d_j\|_2^2 \right). \tag{16}$$

The largest positive contributor is promoted.

**Random-promotion DMSAE.** One eligible stable candidate is selected uniformly at random from Prefix 0. Unlike the reconstruction- and pLDDT/pTM-guided policies, random promotion does not rank candidates by an attribution score.

**Ordered SAE.** The Ordered comparator uses per-example TopK and a weighted objective over individual prefixes (Wang et al., 2025). It has no retained core or promotion mechanism.

**MP-SAE.** MP-SAE repeatedly selects a dictionary atom against the current residual and updates that residual before the next selection (Costa et al., 2025). This is an architecture-level comparator rather than a promotion-policy ablation.




Figure 7: **DMSAEs produce stronger and more consistent pLDDT/pTM attribution for amino acid-associated latents.** For each amino acid, we show attribution scores for latents with cosine similarity $> 0.5$ to its consensus direction. DMSAE variants generally have higher attribution scores than vanilla Matryoshka, Ordered SAE, and MP-SAE. MP-SAE often splits the same amino acid feature across several latents, which can reduce attribution to any single latent. The pLDDT/pTM-guided DMSAE also shows clearer separation between amino acids. These results show that feature alignment, attribution, and fragmentation should be evaluated together.