Interpreting Genomic Language Models using Sparse Autoencoders

Anonymous Authors[1]

Abstract

Genomic language models (gLMs) achieve strong performance across diverse genomic prediction tasks, but their internal biological representations remain poorly understood. Sparse autoencoders (SAEs) have emerged as an interpretability tool in vision and natural language models, yet their applicability to gLMs remains unexplored. We present a systematic study of SAE-based interpretability for gLMs, introducing a diverse benchmark of human genomic annotations and a suite of genome-tailored interpretability metrics. Using Evo2 as a primary case study, we show that SAE features, particularly those from intermediate layers, are more interpretable than raw model embeddings across 42/55 (76%) of our genomic concept evaluations, with 26 of them having an F1 score greater than 0.7. We further find that interpretability depends on SAE training data properties such as evolutionary proximity and context length, with mixed-species and longer-context training improving recovery of human genomic features. Finally, we develop a graph-based representation method to construct a feature atlas that organizes semantically related genomic concepts learned by an SAE, outperforming the baseline approach of using SAE model weights. Our results establish SAEs as a powerful framework for better understanding gLMs, broadening their accessibility and utility for disease-driven genomic analysis.

1. Introduction

Advances in high-throughput sequencing and genomic databases have generated vast biological sequence data, driving the development of machine learning models for DNA analysis (Benegas et al., 2024; van Hilten et al., 2024; Libbrecht & Noble, 2015; Hassan et al., 2022). Among these, genomic language models (gLMs) trained across multiple species have demonstrated a remarkable performance across downstream tasks, such as predicting the effect of nucleotide mutations and identifying regulatory elements such as transcription factor binding sites (Saadat & Fellay, 2024; Sanabria et al., 2024; Benegas et al., 2025; 2024). These results suggest that gLMs may harbor rich representations of DNA grammar.

Despite their success, gLMs remain difficult to interpret, as their internal mechanisms for processing DNA sequences and representing biological concepts are largely opaque (Consens et al., 2025). Moreover, genomics provides a uniquely adversarial setting for interpretation due to long sequence lengths and incomplete ground truth functional annotations. These properties have complicated evaluations of gLMs and limited their adoption in biomedical research, as researchers would struggle to reliably understand and validate model predictions.

Mechanistic interpretability aims to address these limitations by identifying *features*: the fundamental units of representation in neural networks reflecting semantically meaningful properties of the input (Bereska & Gavves, 2024). Isolating such features is difficult due to polysemanticity (Olah et al., 2020), where individual neurons respond to multiple, seemingly unrelated inputs. This behavior is thought to arise from superposition, in which a network represents more features than it has neurons by encoding each feature as a combination of neurons (Bereska & Gavves, 2024).

Sparse autoencoders (SAEs) have recently emerged as a powerful approach for resolving superposition in vision and natural language models (Cunningham et al., 2023; Braun et al., 2024; Pach et al., 2025). By enforcing sparse, overcomplete representations, SAEs disentangle polysemantic neurons and recover interpretable features that align with meaningful concepts.

In this work, we present a systematic study of SAE-based interpretability for DNA language models. We first curate a comprehensive genomic annotation dataset comprised of a diverse range of genomic concepts (such as exons, promoters, enhancers) and where they lie within the human reference genome. We focus our interpretability efforts on human DNA as genomic variation in the human genome underlies disease susceptibility, cancer development, and drug

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

For ICML 2026 reviewers: When reviewing this paper, please follow the policy for the use of LLMs as specified on your Reviewer Console. Note that the assigned LLM policy might differ from the policy you indicated during reviewer registration.

response, making interpretable representations directly relevant to biomedical and clinical applications (Khodadadian et al., 2020). By providing this benchmark, we establish a standardized basis for quantitatively assessing whether SAE features align with known biological concepts.

Using Evo2 (Brixi et al., 2025) as a case study gLM, we analyze SAE features across multiple model layers. We show that different genomic concepts are preferentially encoded at different depths and that intermediate layers yield the most interpretable SAE representations. Across the majority of annotated genomic elements, SAE features exhibit significantly stronger linear alignment with biological concepts than the corresponding raw model embeddings. We further demonstrate that SAE performance depends on SAE training set attributes and its hyperparameters.

Finally, to move beyond feature-level evaluation, we introduce a representation learning algorithm that embeds SAE features into a shared space based on the similarity of their activation patterns. This approach yields a visualizable "feature atlas" that organizes semantically related features and captures the global structure of the SAE dictionary.

In summary, our contributions are as follows:

- We curate a dataset of human genomic annotations and show that SAE features align more strongly with genomic concepts than raw embeddings, identifying intermediate model layers as maximally interpretable
- We find that SAE interpretability depends on training data provenance (such as evolutionary proximity between training genomes and target genomic concepts and sequence context length)
- We propose an algorithm to effectively represent SAE features into a coherent atlas, capturing the scale and structure of the SAE dictionary space and allowing for feature annotation transfer and exploration of unknown genomic patterns

2. Background

2.1. Genomic Language Models (gLMs)

Genomic language models (gLMs) have emerged as large transformer architectures trained on raw genomic sequences, usually through a masked language modeling or a causal language modeling scheme (Benegas et al., 2024). gLMs learn predictive representations of regulatory grammar, functional elements, and evolutionary patterns in DNA. Numerous models that have been proposed, such as Evo2 (Brixi et al., 2025), Nucleotide Transformer (Dalla-Torre et al., 2025), and AlphaGenome (Avsec et al., 2025), represent genomic sequences as tokens, typically as individual nucleotides or short k-mers (substrings of length k from a biological sequence). These tokens are mapped into continuous embedding spaces, similar to word embeddings in natural language models.

2.2. Sparse Autoencoders (SAEs)

SAEs are small neural networks with a single hidden layer that are trained post-hoc on embeddings from an intermediate layer of a pretrained language model (Cunningham et al., 2023). Given an embedding $x \in \mathbb{R}^d$ of dimension d extracted from a pretrained model, an SAE learns an overcomplete dictionary of m features ($m \gg d$) through an encoder-decoder mapping:

$$\hat{x} = W_{\text{dec}}(\sigma(W_{\text{enc}}x + b_{\text{enc}})) + b_{\text{dec}}$$
$$W_{\text{enc}} \in \mathbb{R}^{m\times d}, b_{\text{enc}} \in \mathbb{R}^m, W_{\text{dec}} \in \mathbb{R}^{d\times m}, b_{\text{dec}} \in \mathbb{R}^d$$

where σ is an activation function. Model weights are optimized to minimize reconstruction error while enforcing sparsity in the hidden layer's activations. Sparsity can be enforced in different ways depending on the type of SAE, such as by allowing only a specific number of hidden units to activate per token or by adding an L1 penalty of the hidden activations as a loss term (Bussmann et al., 2024). These attributes allow the SAE to produce a disentangled representation where features are selectively activated.

3. Related Work

3.1. SAEs for Protein Language Models

InterPLM is an approach to decode the internal mechanisms of protein language models (PLMs) such as ESM-2 using SAEs (Simon & Zou, 2025). They find that SAEs decompose dense PLM embeddings into interpretable features which successfully map to coherent protein attributes spanning both structural and sequential characteristics. They further observe that SAE features better capture biological concepts than individual neurons and that these features can be grouped based on shared functional or structural roles. While InterPLM establishes the effectiveness of SAEs for biological interpretability, applying similar approaches to genomic data is more challenging. This is due to the much larger length of genomic sequences and the relative sparsity of functional annotations, reflecting the limited domain knowledge for the majority of the genome (Brown & Lad, 2019; Oprea et al., 2018).

3.2. Evo2

Evo2 is currently the largest fully open source genomic language model and was trained on trillions of nucleotide sequences from all domains of life, spanning roughly 128,000 genomes. (Brixi et al., 2025; Consens et al., 2025). Evo2 adopts the Hyena Convolutional architecture, which enables

Confidential reviewer copy. This manuscript is under double-blind review by ICML 2026. Unauthorized sharing, redistribution, or disclosure is strictly prohibited.

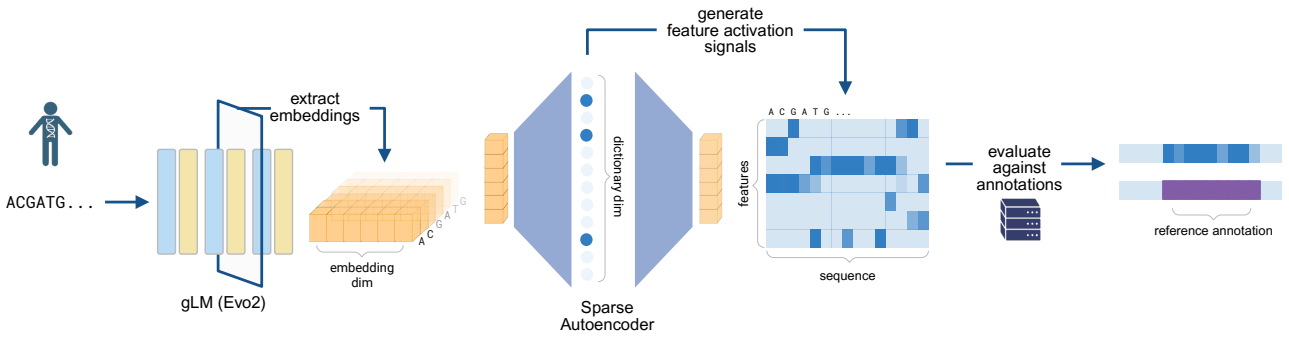

Figure 1. **Summary of workflow.** DNA sequences are passed into the gLM, and embeddings are extracted. These embeddings are used to train sparse autoencoders (SAEs). After training an SAE, test sequences' embeddings are converted into feature activation signals. These signals are compared against known genomic annotations using our evaluation metrics. (Created with BioRender.com)

it to efficiently handle an exceptionally long context window of up to one million tokens, which are defined at a single nucleotide (single base pair) resolution. In collaboration with Goodfire AI, the authors discovered that SAEs are capable of synthesizing complex, high-level features that correspond to known biological properties across DNA, RNA, protein, and organism levels. While this initial work provided compelling evidence for the adoption of SAEs, the scope of their analysis was limited as it focused on a fairly limited set of genomic features and a single model layer.

4. Genomic Annotations for SAE Benchmarking

We compiled a comprehensive evaluation suite spanning gene structure elements (promoters, exons, introns, UTRs, and coding sequences), regulatory regions (candidate cis regulatory elements, enhancers, and CpG islands), repeat elements, and evolutionarily conserved regions. Gene structure annotations were derived from GENCODE v49 (Mudge et al., 2024) and MANE v1.4 (Morales et al., 2022). Regulatory annotations were obtained from the ENCODE cis Regulatory Elements Registry V4 (ENCODE Project Consortium et al., 2020) and SEdb (Wang et al., 2023). The suite further integrates protein coding and non coding biotypes, including lncRNA, miRNA, tRNA, snoRNA, and piRNA, alongside transposable and repetitive element classes and evolutionary conservation elements. For each annotation category, genomic coordinates were extracted and corresponding DNA sequences were obtained from the reference genome, enabling systematic evaluation of the correspondence between SAE features and known functional genomic elements.

For benchmarking SAEs against these annotations, we used chromosome 22 as a heldout test set and defined 55 genomic concepts, evaluated in strand-specific and/or strand-invariant settings according to biological context. Various sequences of length 10kb were sampled from chromosome 22 to construct balanced evaluation sets for each concept. Additional details are provided in Appendix A.

5. Metrics for SAE Evaluation

While there exists several SAE evaluation metrics that were designed to match SAE-derived features with known concepts (Karvonen et al., 2025; Gurnee et al., 2023), the majority of these metrics were designed for natural language or vision tasks and do not immediately translate for working with biological sequences. To support our evaluation of the genomic annotations, we present a set of metrics tailored towards identifying a wide variety of genomic concepts. These metrics are used to efficiently (i) identify the most interpretable features for a given annotation and (ii) quantify the strength of feature–annotation correspondence. Throughout, we report the performance of a single best feature for each annotation, reflecting the goal of learning monosemantic latent features. Full details can be found in Appendix B.

5.1. Nucleotide Precision and Domain Recall F1 Score

We adopt the modified F1 score proposed by Simon and Zou (Simon & Zou, 2025), which evaluates precision at the nucleotide level and recall at the domain level. The precision component measures the fraction of a feature's activated nucleotides that lie within annotated instances of a given concept. The recall component measures the fraction of annotated concept instances that are detected by the feature. Here, a domain corresponds to a contiguous genomic segment labeled positive for a given annotation. Feature activation signals are binarized using a fixed threshold ($\tau = 0$), and a domain is considered detected if the feature activates at least once at any position within that segment.

For each annotation, we report the maximum F1 score achieved by any single feature. This metric is well suited for

Confidential reviewer copy. This manuscript is under double-blind review by ICML 2026. Unauthorized sharing, redistribution, or disclosure is strictly prohibited.

features with sparse or localized activations, which are common for SAE-derived features that respond to short sequence motifs rather than entire annotated segments. However, it relies on a selected threshold for activation, discarding potentially important variation in feature activation magnitude across a sequence.

5.2. Sparse Probing

To assess whether individual features linearly predict genomic annotations, we employ sparse probing (Gurnee et al., 2023). Features are ranked by their mean activation difference between annotated and unannotated positions, and the top k candidates are evaluated using univariate logistic regression models, referred to as probes.

Unlike prior work that combines a few features to train a single probe, we probe one feature at a time to align with the interpretability objective of having a one-to-one correspondence between SAE-derived features and concepts.

For each annotation, we report the highest Area Under the Precision-Recall Curve (AUPRC) achieved by a probe. The sparse probing metric is beneficial as it directly leverages SAE feature activation magnitudes but has a bias towards features with consistent domain-wide responses.

5.3. Sparse Event Activation

Certain genomic events, such as transcription start sites, occur at very few genomic positions and are poorly represented by domain-based metrics. To recover features associated with such rare events, we propose a sparse event activation score that loosens the constraint of a feature activating exactly at sparse events. This is done by measuring feature signal enrichment within a fixed window surrounding sparse event sites relative to remaining background regions. Feature activations are smoothed using a Gaussian-weighted window, and enrichment is computed as a log-ratio between event-centered and background activations.

For each annotation, we report the feature with the highest enrichment score along with its F1 score. This metric is effective in identifying features linked to rare or pointwise genomic events but also depends on the choice of window size and smoothing parameters.

6. Feature Representation via Activation-Based Similarity

While metrics quantify how well individual features align with specific annotations, they do not reveal higher-order relationships among SAE-derived features. Understanding these relationships is important for identifying coherent groups of features and producing visualizable representations that summarize the feature space learned by the SAE. Prior work has represented SAE features using decoder weight similarity (Simon & Zou, 2025). However, this primarily reflects reconstruction behavior of the SAE and does not necessarily correspond to similarity in feature activation patterns. In an ideal embedding of SAE features, features that activate in similar genomic contexts should be placed close together.

Based on this goal, we first compute feature activation signals over a 500kb representative genomic sequence. This sequence should be sufficiently long to maximize the number and diversity of observed genomic events (and consequently SAE feature activations). Details on choosing this sequence's length can be found in Figure 8 in the Appendix. Using these signals, we apply a smoothing convolution to capture spatially proximal activations. After a downsampling step, we identify approximate nearest neighbors (ANN) for each feature. For each pair of neighboring features, we compute a continuous Tanimoto coefficient on the downsampled signals. The Tanimoto coefficient is a metric that extends Jaccard similarity, and has been shown to work well with sparse data, and has the advantage of taking into consideration both the angle and the length of vectors in its computation (Anastasiu & Karypis). The neighbors and calculated Tanimoto coefficients are used to initialize the nodes and edges of a graph, where each SAE-derived feature is a node and connecting edges represent activation pattern similarity. Finally, node embeddings are learned from this graph using the node2vec algorithm (Grover & Leskovec, 2016). A summary of these steps is provided in Algorithm 1 with further details provided in Appendix C.

Algorithm 1 Tanimoto / Node2Vec Feature Representation

Input: Feature signals $\mathbf{X} \in \mathbb{R}^{F \times L}$, nearest neighbors k, embedding dimension h
Output: Embedding $\mathbf{Z} \in \mathbb{R}^{F \times h}$
$\mathbf{X}_{\text{smooth}} \leftarrow \text{UniformFilter}(\mathbf{X})$ and downsample
Find k-nearest neighbors for each feature using ANN
Compute Tanimoto coefficient affinity matrix $\mathbf{K}$:
$K_{ij} = \frac{\mathbf{x}_i \mathbf{x}_j}{\|\mathbf{x}_i\|^2 + \|\mathbf{x}_j\|^2 - \mathbf{x}_i \mathbf{x}_j}$ for neighbors, 0 elsewhere
Initialize graph $\mathbf{G}$ using $\mathbf{K}$ as an adjacency matrix
$\mathbf{Z} \leftarrow \text{node2vec}(\mathbf{G}, h)$
return Z

7. Results

For our experiments, we follow the workflow outlined in Figure 1. We begin by pooling a set of sequences to train our SAE, pass them through the gLM, and extract embeddings at an intermediate layer. Using the embeddings, we train a BatchTopK sparse autoencoder (Bussmann et al., 2024). Details on the SAE hyperparameters can be found in Appendix D. We then process our held-out testing sequences by ob-

Confidential reviewer copy. This manuscript is under double-blind review by ICML 2026. Unauthorized sharing, redistribution, or disclosure is strictly prohibited.

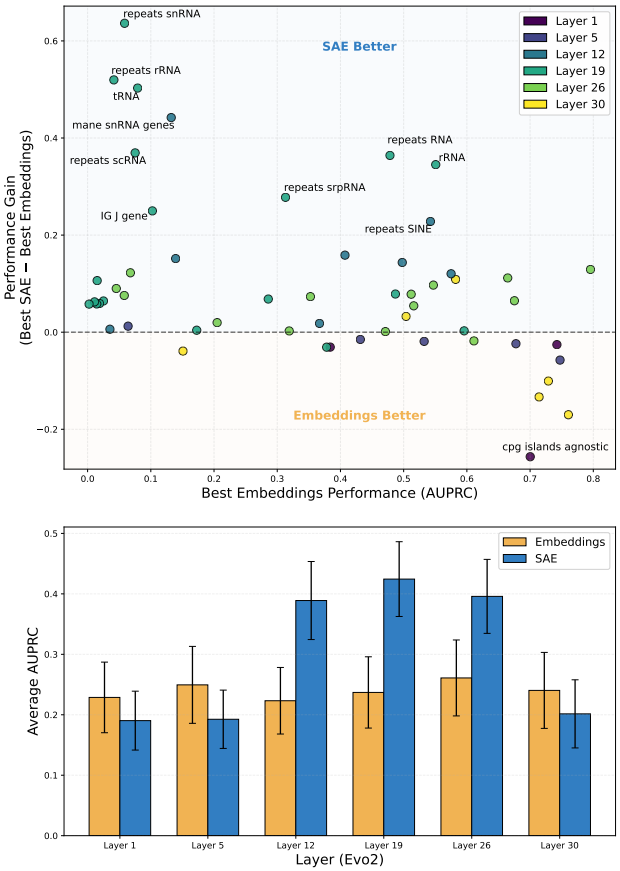

Figure 2. **Comparison of SAE features and Evo2 embeddings aligning with genomic annotations.** *(Top)* Each point corresponds to an annotation and is colored by the Evo2 layer that yielded the highest performance for that concept. Gain is the difference between the best SAE probe AUPRC and the best embeddings probe AUPRC (across all six evaluated layers). *(Bottom)* Average performance of embeddings and SAEs of all benchmarked concepts at each layer, with standard deviation across concepts.

taining embeddings from the same layer, which are passed through the trained SAE to generate feature activation signals, which represent the response of each SAE-derived feature across the input sequence. We benchmark these signals against our available genomic annotations.

7.1. Intermediate-layer SAE features outperform embeddings for interpreting genomic concepts

We aimed to explore how different genomic concepts are captured in the gLM and whether SAEs can better recover these concepts from the model's representations. To do this, we trained SAEs on six layers of Evo2 and three layers of Nucleotide Transformer (NT) v3 as an additional comparison (Boshar et al., 2025). We evaluated whether the resulting feature activation signals align more closely with genomic concepts than the embeddings themselves. To enable a fair comparison, we use the sparse probing framework in which each probe consists of a single dimension: either one embedding neuron or one SAE feature.

We observed a clear layer-dependent pattern in both gLMs. Embeddings marginally outperform SAE features at early and late layers, while SAEs achieve superior performance in the intermediate layers (Figure 2). For each genomic concept, we also report the layer and representation type (embeddings vs. SAE features) of the best-performing probe. Notably, for Evo2, 42 of the 55 genomic concepts (76%) were best captured by SAEs; for NT, 30 of the 55 (55%). For Evo2, we also observe that 26 evaluations achieve a modified F1 score value greater than 0.7. Selected results are presented in Table 1, with the complete set provided in Appendix E; full NT results are reported in Appendix F.

Even for concepts where embeddings yielded better probe performance, the observed gains were generally small (Figure 2). Among concepts with negative gain, the median magnitude was 0.031 for Evo2 and 0.060 for Nucleotide Transformer, compared to median positive gains of 0.084 and 0.128, respectively.

As expected with the probe metric, the annotations with rare events such as TSS (transcription start site), start and stop codons, and miRNA have a low AUPRC. We investigated these annotations further with the sparse event activation metric, and report the results in Appendix G. While most of the sparse annotations could not be perfectly recovered in the SAEs, we still observed a similar pattern of concepts being best captured in intermediate layers.

7.2. Training set attributes influence SAE-derived feature alignment with human genomic concepts

We investigated how the choice of training data influences the ability of SAEs to capture biologically meaningful genomic concepts, focusing on the axes of source genome and

Table 1. **Probe evaluation of selected genomic concepts.** The winner is selected based on whether SAEs or embeddings (Evo2) yielded better probe performance, defined as the maximum AUPRC across evaluated layers.

Annotation	Winner	AUPRC	Layer
cpg islands	Embeddings	0.700	1
crispr targets	Embeddings	0.678	5
IG V gene	SAE	0.644	26
mane introns	SAE	0.776	26
miRNA	SAE	0.089	19
repeats Retroposon	SAE	0.925	26
repeats SINE	SAE	0.770	12
rRNA	SAE	0.896	19
tRNA	SAE	0.582	19
typical enhancers	SAE	0.536	30

Confidential reviewer copy. This manuscript is under double-blind review by ICML 2026. Unauthorized sharing, redistribution, or disclosure is strictly prohibited.

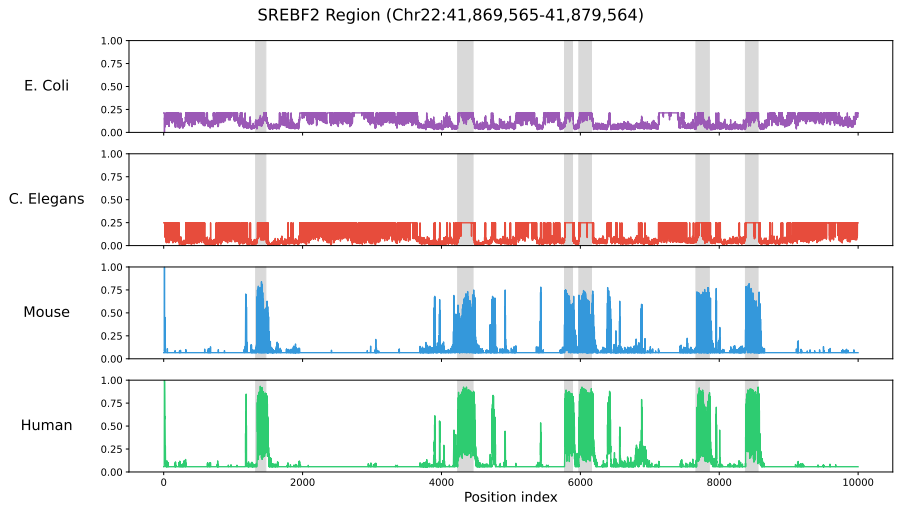

Figure 3. **Example of probe predictions of exons on a selected human gene.** SAEs were trained on E. Coli, C. Elegans, Mouse, and Human genomes and evaluated on a heldout set of human genes. Sparse probes were trained to identify exons using a single SAE-derived feature. Here, at each nucleotide in the sequence, the probe predictions represent across this gene, the probability that each nucleotide is in an exon. MANE annotated exons are highlighted in grey.

context length. To evaluate the effect of source genome, we assessed the ability of SAE feature signals to recover exons (genetic elements that arose in eukaryotic lineages) in human DNA. We trained separate SAEs on Evo2 embeddings of 5 million total base pairs from four genomes of increasing evolutionary proximity to humans: E. coli, C. elegans, mouse, and human. SAEs were evaluated using sparse probing and F1 score. We also trained an SAE on a mixed-species dataset (also 5 million total base pairs with equal representation across the four species). Each SAE was then evaluated on a held-out set of human genes comprising both non-conserved and highly-conserved homologs (Appendix H). An example of probe predictions for each of the 4 single-genome SAEs is shown in Figure 3.

Figure 4 shows that while some exons are captured consistently across all SAEs, many exons are only accurately identified when the SAE is trained on genomes more closely related to humans. This suggests that evolutionary proximity of the training sequences impacts the SAE's ability to encode human-relevant genomic features. However, there still persists a large variability in feature-concept alignment between the evaluated genes, which could reflect heterogeneity in exon conservation, alternative splicing patterns, or uncaptured regulatory signals.

Interestingly, the mixed-species SAE achieves a remarkably strong performance, even slightly outperforming the human-only SAE despite having been trained on fewer human DNA sequences. We think that allowing the SAE to see exons' structure across evolution may help the SAE detect a more diverse set of exon-related patterns that encourage better generalizability.

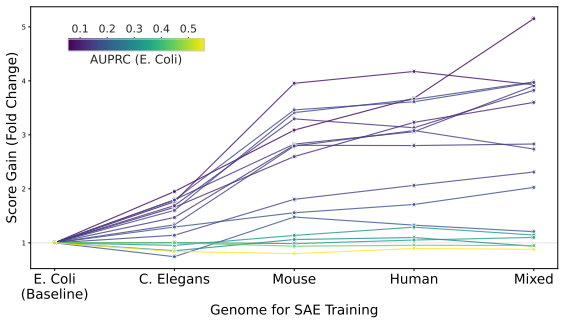

Figure 4. **Performance gain of exon interpretability across training genomes.** Each line represents a different evaluated DNA region. The y-axis represents the relative gain (defined as fold change) in AUPRC performance between an SAE trained on each genome and the E. Coli-trained SAE baseline. Lines are colored by the baseline AUPRC performance.

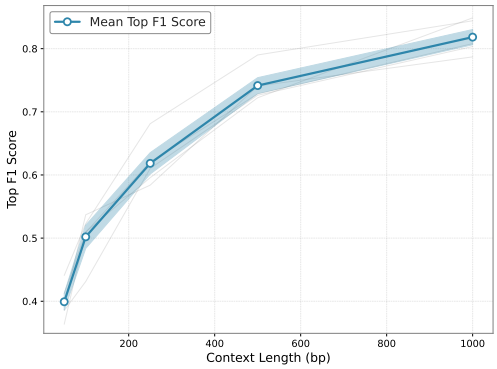

Figure 5. **Effect of context size on F1 score of SAEs.** SAEs were trained with varying context lengths for embeddings and evaluated on exon annotations using the F1 score metric.

Confidential reviewer copy. This manuscript is under double-blind review by ICML 2026. Unauthorized sharing, redistribution, or disclosure is strictly prohibited.

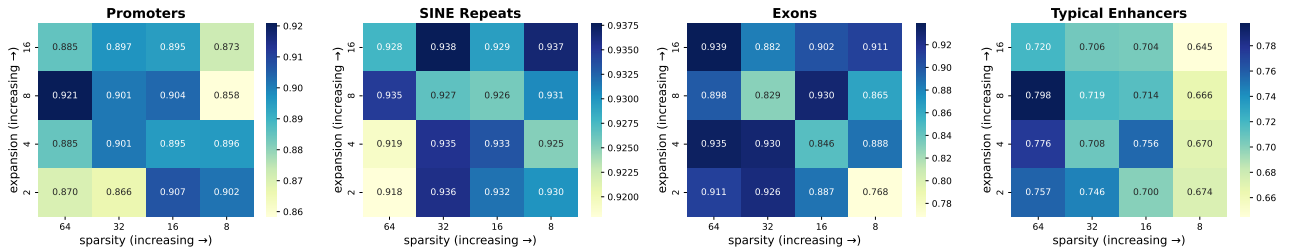

Figure 6. **Hyperparameter effects on interpretability.** SAEs were trained across different dictionary sizes (based on different expansion factors) and sparsities (defined as choice of k for BatchTopK SAEs) and evaluated with the F1 score metric.

We also further explored the effect of context length of sequences used to generate embeddings for training SAEs (see Appendix I for details). Figure 5 demonstrates that reliable identification of exons benefits from training on embeddings from sequences with longer context windows.

These results underscore the importance of carefully selecting appropriate training sequences for SAEs, supporting previous observations of SAE sensitivity to training data (Kissane et al., 2024). Because we are concerned in identifying human-relevant features, these results justify prioritizing human DNA for SAE training when applying SAEs for human-centric tasks.

7.3. Sparsity and dictionary size have mixed effects on SAE feature quality

Two key hyperparameters in training BatchTopK SAEs are the dictionary size (the number of features to extract) and the sparsity level (K for TopK and BatchTopK sparse autoencoders) (Bussmann et al., 2024; Gao et al., 2024). We conducted a grid search across this hyperparameter space to identify patterns influencing feature performance. While performance fluctuates across settings, the effects of these hyperparameters differ depending on the genomic concept of interest (Figure 6). In other words, no single combination of dictionary size and sparsity consistently yields optimal performance across all genomic concepts. Furthermore, it appears some concepts (i.e. promoters, enhancers) are more sensitive to hyperparameter changes than others. This suggests that SAE hyperparameters may need to be adjusted to fit specific biological features or analysis goals, rather than adopting a "one-size-fits-all" configuration.

7.4. Representations of SAE-derived features based on signal patterns

After training and tuning an SAE, a natural next step is identifying higher-order relationships among SAE-derived features. Using our Tanimoto / Node2Vec algorithm (Algorithm 1), we seek to uncover coherent feature groupings and provide interpretable representations of the learned feature space. To test the robustness of our algorithm for represent-

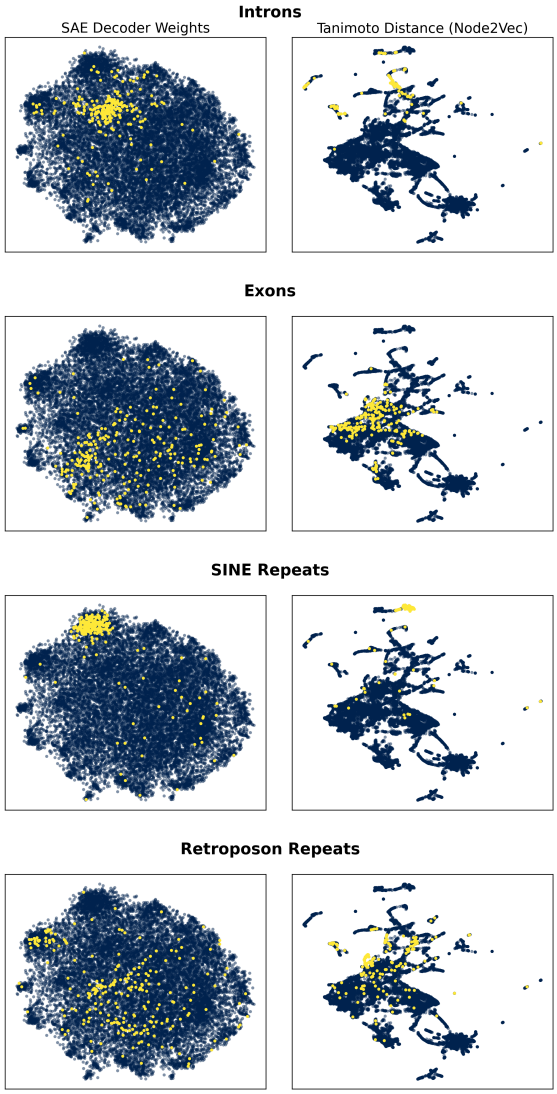

Figure 7. **UMAP representations highlighting high performing features.** Candidate features for four genomic concepts (defined as having high F1 scores) are highlighted in the representation generated using SAE decoder weights and in the representation generated using the proposed method.

Confidential reviewer copy. This manuscript is under double-blind review by ICML 2026. Unauthorized sharing, redistribution, or disclosure is strictly prohibited.

Table 2. **kNN precision evaluation of representation of high-performing features.** Evaluation compares the SAE weights method with the proposed method (Tanimoto + Node2Vec) and was done with varying levels of local resolution ($k = \{10, 25, 50\}$). A higher kNN precision represents a better organization of features for that concept.

	SAE Weights			Tanimoto + Node2Vec (ours)		
Feature	k=10	k=25	k=50	k=10	k=25	k=50
Introns	0.469	0.442	0.413	0.663	0.626	0.598
Exons	0.121	0.116	0.107	0.135	0.129	0.124
SINE Repeats	0.595	0.554	0.510	0.735	0.706	0.689
Retroposon Repeats	0.211	0.198	0.172	0.340	0.312	0.269

ing SAE-derived features, we benchmarked it against the method proposed by Simon and Zou for one of our trained SAEs. We evaluate representations both qualitatively and quantitatively. Qualitatively, we used UMAP visualizations, highlighting the high-performing features, defined by being above the 98.5th percentile of F1 scores for the respective concept. Quantitatively, we calculated a k-nearest neighbors (kNN) precision metric, defined as

$$\frac{1}{|S_\tau|} \sum_{i \in S_\tau} \frac{1}{k} \sum_{j \in \mathcal{N}_k(i)} \mathbb{I}[j \in S_\tau]$$

where S_τ represents the set of high-performing features and $\mathcal{N}_k(i)$ are the k-nearest neighbors to feature i in the embedding space. The metric measures the fraction of neighbors of high-performing features that are also high-performing.

We observe that qualitatively, both methods are able to distinctly group features associated with some tested annotations (Figure 7). However, using the kNN precision evaluation, our method significantly outperforms the SAE decoder weights approach across multiple different features and different settings of k (Table 2). These results indicate that, while both methods are able to represent features based on signal patterns, our method provides a tighter and cleaner organization of SAE-derived features compared to simply using the SAE weights.

Limitations

While our study demonstrates the promise of sparse autoencoders for interpreting DNA language models, several limitations remain.

First, our evaluation relies on existing genomic annotations, which are inherently incomplete and heterogeneous in quality. Many functional genomic elements have strong cell type, developmental stage, or environmental dependencies. These dependencies could explain why a candidate feature may be underperforming for its corresponding annotation in a way that is unrelated to the underlying gLM it was extracted from.

Second, the computational cost of SAE-based interpretability is non-trivial. Training SAEs on large-scale gLM embeddings requires significant memory and compute resources, particularly when evaluating them at scale. Additionally, our feature representation algorithm, which relies on long-range activation patterns and approximate nearest neighbor searches, incurs significant overhead when scaling to SAEs with very large dictionary dimensions.

Conclusions

In this work, we conducted a systematic study of sparse autoencoders as an interpretability framework for DNA language models. Using Evo2 as a case study, we showed that SAE-derived features align more strongly with known genomic elements than raw embeddings, particularly in intermediate model layers. We further demonstrate that SAE interpretability depends sensitively on training set attributes and hyperparameters.

We additionally introduced an activation-based representation method that organizes SAE-derived features into a structured feature atlas. The method successfully groups related SAE-derived features, and in the future, we hope to assess how this method can be used in the exploration of uncharacterized features that do not closely correspond to known genomic concepts. We think that these features may reflect noise, rare events, gLM or SAE model artifacts, or genuinely novel genomic signals that fall outside current biological knowledge. Distinguishing between these possibilities, however, remains a difficult problem.

Together, these results show how sparse autoencoders can be a practical and effective tool for interpreting genomic language models, despite the open challenges related to annotation coverage, feature resolution, and scalability. We view this work as a foundation for future genomic interpretability studies, including the application of gLMs for genetic variant analysis and other disease-relevant applications.

Impact Statement

This paper advances the interpretability of biological foundation models. By extracting and identifying understand-

Confidential reviewer copy. This manuscript is under double-blind review by ICML 2026. Unauthorized sharing, redistribution, or disclosure is strictly prohibited.

able features in gLMs and connecting them with known biological concepts, we establish a transparent framework through which we can begin to explore how different disease-associated mutations and genomic characteristics might modulate gLM behavior. However, a potential negative impact is that some extracted features may reflect dataset bias or modeling artifacts, which could lead to misleading and inaccurate biological interpretations if not carefully validated.

References

Anastasiu, D. C. and Karypis, G. Efficient Identification of Tanimoto Nearest Neighbors.

Avsec, Ž., Latysheva, N., Cheng, J., Novati, G., Taylor, K. R., Ward, T., Bycroft, C., Nicolaisen, L., Arvaniti, E., Pan, J., Thomas, R., Dutordoir, V., Perino, M., De, S., Karollus, A., Gayoso, A., Sargeant, T., Mottram, A., Wong, L. H., Drotár, P., Kosiorek, A., Senior, A., Tanburn, R., Applebaum, T., Basu, S., Hassabis, D., and Kohli, P. AlphaGenome: Advancing regulatory variant effect prediction with a unified DNA sequence model, July 2025. ISSN 2692-8205.

Benegas, G., Ye, C., Albors, C., Li, J. C., and Song, Y. S. Genomic Language Models: Opportunities and Challenges. *ArXiv*, pp. arXiv:2407.11435v2, September 2024. ISSN 2331-8422.

Benegas, G., Albors, C., Aw, A. J., Ye, C., and Song, Y. S. A DNA language model based on multispecies alignment predicts the effects of genome-wide variants. *Nature Biotechnology*, pp. 1–6, January 2025. ISSN 1546-1696. doi: 10.1038/s41587-024-02511-w.

Bereska, L. and Gavves, E. Mechanistic Interpretability for AI Safety – A Review, April 2024.

Boshar, S., Evans, B., Tang, Z., Picard, A., Adel, Y., Lorbeer, F. K., Rajesh, C., Karch, T., Sidbon, S., Emms, D., Mendoza-Revilla, J., Al-Ani, F., Seitz, E., Schiff, Y., Bornachot, Y., Hernandez, A., Lopez, M., Laterre, A., Beguir, K., Koo, P., Kuleshov, V., Stark, A., de Almeida, B. P., and Pierrot, T. A foundational model for joint sequence-function multi-species modeling at scale for long-range genomic prediction, December 2025. ISSN 2692-8205.

Braun, D., Taylor, J., Goldowsky-Dill, N., and Sharkey, L. Identifying Functionally Important Features with End-to-End Sparse Dictionary Learning, May 2024.

Brixi, G., Durrant, M. G., Ku, J., Poli, M., Brockman, G., Chang, D., Gonzalez, G. A., King, S. H., Li, D. B., Merchant, A. T., Naghipourfar, M., Nguyen, E., Ricci-Tam, C., Romero, D. W., Sun, G., Taghibakshi, A., Vorontsov, A., Yang, B., Deng, M., Gorton, L., Nguyen, N., Wang, N. K., Adams, E., Baccus, S. A., Dillmann, S., Ermon, S., Guo, D., Ilango, R., Janik, K., Lu, A. X., Mehta, R., Mofrad, M. R. K., Ng, M. Y., Pannu, J., Ré, C., Schmok, J. C., John, J. S., Sullivan, J., Zhu, K., Zynda, G., Balsam, D., Collison, P., Costa, A. B., Hernandez-Boussard, T., Ho, E., Liu, M.-Y., McGrath, T., Powell, K., Burke, D. P., Goodarzi, H., Hsu, P. D., and Hie, B. L. Genome modeling and design across all domains of life with Evo 2, February 2025.

Brown, S. D. M. and Lad, H. V. The dark genome and pleiotropy: Challenges for precision medicine. *Mammalian Genome*, 30(7):212–216, August 2019. ISSN 1432-1777. doi: 10.1007/s00335-019-09813-4.

Bussmann, B., Leask, P., and Nanda, N. BatchTopK Sparse Autoencoders, December 2024.

Chanin, D., Wilken-Smith, J., Dulka, T., Bhatnagar, H., Golechha, S., and Bloom, J. A is for Absorption: Studying Feature Splitting and Absorption in Sparse Autoencoders. 2024. doi: 10.48550/ARXIV.2409.14507.

Consens, M. E., Li, B., Poetsch, A. R., and Gilbert, S. Genomic language models could transform medicine but not yet. *npj Digital Medicine*, 8(1):212, April 2025. ISSN 2398-6352. doi: 10.1038/s41746-025-01603-4.

Cunningham, H., Ewart, A., Riggs, L., Huben, R., and Sharkey, L. Sparse Autoencoders Find Highly Interpretable Features in Language Models. 2023. doi: 10.48550/ARXIV.2309.08600.

Dalla-Torre, H., Gonzalez, L., Mendoza-Revilla, J., Lopez Carranza, N., Grzywaczewski, A. H., Oteri, F., Dallago, C., Trop, E., de Almeida, B. P., Sirelkhatim, H., Richard, G., Skwark, M., Beguir, K., Lopez, M., and Pierrot, T. Nucleotide Transformer: Building and evaluating robust foundation models for human genomics. *Nature Methods*, 22(2):287–297, February 2025. ISSN 1548-7105. doi: 10.1038/s41592-024-02523-z.

ENCODE Project Consortium, Moore, J. E., Purcaro, M. J., Pratt, H. E., Epstein, C. B., Shoresh, N., Adrian, J., Kawli, T., Davis, C. A., Dobin, A., Kaul, R., Halow, J., Van Nostrand, E. L., Freese, P., Gorkin, D. U., Shen, Y., He, Y., Mackiewicz, M., Pauli-Behn, F., Williams, B. A., Mortazavi, A., Keller, C. A., Zhang, X.-O., Elhajjajy, S. I., Huey, J., Dickel, D. E., Snetkova, V., Wei, X., Wang, X., Rivera-Mulia, J. C., Rozowsky, J., Zhang, J., Chhetri, S. B., Zhang, J., Victorsen, A., White, K. P., Visel, A., Yeo, G. W., Burge, C. B., Lécuyer, E., Gilbert, D. M., Dekker, J., Rinn, J., Mendenhall, E. M., Ecker, J. R., Kellis, M., Klein, R. J., Noble, W. S., Kundaje, A., Guigó, R., Farnham, P. J., Cherry, J. M., Myers, R. M., Ren, B., Graveley, B. R., Gerstein, M. B., Pennacchio, L. A., Snyder, M. P., Bernstein, B. E., Wold,

Confidential reviewer copy. This manuscript is under double-blind review by ICML 2026. Unauthorized sharing, redistribution, or disclosure is strictly prohibited.

B., Hardison, R. C., Gingeras, T. R., Stamatoyannopoulos, J. A., and Weng, Z. Expanded encyclopaedias of DNA elements in the human and mouse genomes. *Nature*, 583(7818):699–710, July 2020. ISSN 1476-4687. doi: 10.1038/s41586-020-2493-4.

Gao, L., la Tour, T. D., Tillman, H., Goh, G., Troll, R., Radford, A., Sutskever, I., Leike, J., and Wu, J. Scaling and evaluating sparse autoencoders, June 2024.

Grover, A. and Leskovec, J. Node2vec: Scalable Feature Learning for Networks, July 2016.

Gurnee, W., Nanda, N., Pauly, M., Harvey, K., Troitskii, D., and Bertsimas, D. Finding Neurons in a Haystack: Case Studies with Sparse Probing, June 2023.

Hassan, M., Awan, F. M., Naz, A., deAndrés-Galiana, E. J., Alvarez, O., Cernea, A., Fernández-Brillet, L., Fernández-Martínez, J. L., and Kloczkowski, A. Innovations in Genomics and Big Data Analytics for Personalized Medicine and Health Care: A Review. *International Journal of Molecular Sciences*, 23(9):4645, January 2022. ISSN 1422-0067. doi: 10.3390/ijms23094645.

Karvonen, A., Rager, C., Lin, J., Tigges, C., Bloom, J., Chanin, D., Lau, Y.-T., Farrell, E., McDougall, C., Ayonrinde, K., Till, D., Wearden, M., Conmy, A., Marks, S., and Nanda, N. SAEBench: A Comprehensive Benchmark for Sparse Autoencoders in Language Model Interpretability, June 2025.

Khodadadian, A., Darzi, S., Haghi-Daredeh, S., Sadat Eshaghi, F., Babakhanzadeh, E., Mirabutalebi, S. H., and Nazari, M. Genomics and Transcriptomics: The Powerful Technologies in Precision Medicine. *International Journal of General Medicine*, Volume 13:627–640, September 2020. ISSN 1178-7074. doi: 10.2147/IJGM.S249970.

Kissane, C., robertzk, Nanda, N., Conmy, A., robertzk, Nanda, N., and Conmy, A. SAEs are highly dataset dependent: A case study on the refusal direction. November 2024.

Libbrecht, M. W. and Noble, W. S. Machine learning applications in genetics and genomics. *Nature Reviews Genetics*, 16(6):321–332, June 2015. ISSN 1471-0064. doi: 10.1038/nrg3920.

Morales, J., Pujar, S., Loveland, J. E., Astashyn, A., Bennett, R., Berry, A., Cox, E., Davidson, C., Ermolaeva, O., Farrell, C. M., Fatima, R., Gil, L., Goldfarb, T., Gonzalez, J. M., Haddad, D., Hardy, M., Hunt, T., Jackson, J., Joardar, V. S., Kay, M., Kodali, V. K., McGarvey, K. M., McMahon, A., Mudge, J. M., Murphy, D. N., Murphy, M. R., Rajput, B., Rangwala, S. H., Riddick, L. D., Thibaud-Nissen, F., Threadgold, G., Vatsan, A. R., Wallin, C., Webb, D., Flicek, P., Birney, E., Pruitt, K. D., Frankish, A., Cunningham, F., and Murphy, T. D. A joint NCBI and EMBL-EBI transcript set for clinical genomics and research. *Nature*, 604(7905):310–315, April 2022. ISSN 1476-4687. doi: 10.1038/s41586-022-04558-8.

Mudge, J. M., Carbonell-Sala, S., Diekhans, M., Martinez, J. G., Hunt, T., Jungreis, I., Loveland, J. E., Arnan, C., Barnes, I., Bennett, R., Berry, A., Bignell, A., Cerdán-Vélez, D., Cochran, K., Cortés, L. T., Davidson, C., Donaldson, S., Dursun, C., Fatima, R., Hardy, M., Hebbar, P., Hollis, Z., James, B. T., Jiang, Y., Johnson, R., Kaur, G., Kay, M., Mangan, R. J., Maquedano, M., Gómez, L. M., Mathlouthi, N., Merritt, R., Ni, P., Palumbo, E., Perteghella, T., Pozo, F., Raj, S., Sisu, C., Steed, E., Sumathipala, D., Suner, M.-M., Uszczynska-Ratajczak, B., Wass, E., Yang, Y. T., Zhang, D., Finn, R. D., Gerstein, M., Guigó, R., Hubbard, T. J. P., Kellis, M., Kundaje, A., Paten, B., Tress, M. L., Birney, E., Martin, F. J., and Frankish, A. GENCODE 2025: Reference gene annotation for human and mouse. *Nucleic Acids Research*, 53(D1):D966–D975, November 2024. ISSN 0305-1048. doi: 10.1093/nar/gkae1078.

Olah, C., Cammarata, N., Schubert, L., Goh, G., Petrov, M., and Carter, S. Zoom In: An Introduction to Circuits. *Distill*, 5(3):10.23915/distill.00024.001, March 2020. ISSN 2476-0757. doi: 10.23915/distill.00024.001.

Oprea, T. I., Bologa, C. G., Brunak, S., Campbell, A., Gan, G. N., Gaulton, A., Gomez, S. M., Guha, R., Hersey, A., Holmes, J., Jadhav, A., Jensen, L. J., Johnson, G. L., Karlson, A., Leach, A. R., Ma'ayan, A., Malovannaya, A., Mani, S., Mathias, S. L., McManus, M. T., Meehan, T. F., von Mering, C., Muthas, D., Nguyen, D.-T., Overington, J. P., Papadatos, G., Qin, J., Reich, C., Roth, B. L., Schürer, S. C., Simeonov, A., Sklar, L. A., Southall, N., Tomita, S., Tudose, I., Ursu, O., Vidović, D., Waller, A., Westergaard, D., Yang, J. J., and Zahoránszky-Köhalmi, G. Unexplored therapeutic opportunities in the human genome. *Nature Reviews Drug Discovery*, 17(5):317–332, May 2018. ISSN 1474-1784. doi: 10.1038/nrd.2018.14.

Pach, M., Karthik, S., Bouniot, Q., Belongie, S., and Akata, Z. Sparse Autoencoders Learn Monosemantic Features in Vision-Language Models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, October 2025.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances*

Confidential reviewer copy. This manuscript is under double-blind review by ICML 2026. Unauthorized sharing, redistribution, or disclosure is strictly prohibited.

in Neural Information Processing Systems, volume 32. Curran Associates, Inc., 2019.

Perez, G., Barber, G. P., Benet-Pages, A., Casper, J., Clawson, H., Diekhans, M., Fischer, C., Gonzalez, J. N., Hinrichs, A. S., Lee, C. M., Nassar, L. R., Raney, B. J., Speir, M. L., van Baren, M. J., Vaske, C. J., Haussler, D., Kent, W. J., and Haeussler, M. The UCSC Genome Browser database: 2025 update. *Nucleic Acids Research*, 53(D1): D1243–D1249, January 2025. ISSN 1362-4962. doi: 10.1093/nar/gkae974.

Saadat, A. and Fellay, J. DNA Language Model and Interpretable Graph Neural Network Identify Genes and Pathways Involved in Rare Diseases. In Edwards, C., Wang, Q., Li, M., Zhao, L., Hope, T., and Ji, H. (eds.), *Proceedings of the 1st Workshop on Language + Molecules (L+M 2024)*, pp. 103–115, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.langmol-1.13.

Sanabria, M., Hirsch, J., Joubert, P. M., and Poetsch, A. R. DNA language model GROVER learns sequence context in the human genome. *Nature Machine Intelligence*, 6(8): 911–923, August 2024. ISSN 2522-5839. doi: 10.1038/s42256-024-00872-0.

Simon, E. and Zou, J. InterPLM: Discovering interpretable features in protein language models via sparse autoencoders. *Nature Methods*, pp. 1–11, September 2025. ISSN 1548-7105. doi: 10.1038/s41592-025-02836-7.

Tarailo-Graovac, M. and Chen, N. Using RepeatMasker to identify repetitive elements in genomic sequences. *Current Protocols in Bioinformatics*, Chapter 4:4.10.1–4.10.14, March 2009. ISSN 1934-340X. doi: 10.1002/0471250953.bi0410s25.

van Hilten, A., Katz, S., Saccenti, E., Niessen, W. J., and Roshchupkin, G. V. Designing interpretable deep learning applications for functional genomics: A quantitative analysis. *Briefings in Bioinformatics*, 25(5):bbae449, September 2024. ISSN 1477-4054. doi: 10.1093/bib/bbae449.

Wang, J., Zhang, P., Lu, Y., Li, Y., Zheng, Y., Kan, Y., Chen, R., and He, S. piRBase: A comprehensive database of piRNA sequences. *Nucleic Acids Research*, 47(D1): D175–D180, January 2019. ISSN 1362-4962. doi: 10.1093/nar/gky1043.

Wang, Y., Song, C., Zhao, J., Zhang, Y., Zhao, X., Feng, C., Zhang, G., Zhu, J., Wang, F., Qian, F., Zhou, L., Zhang, J., Bai, X., Ai, B., Liu, X., Wang, Q., and Li, C. SEdb 2.0: A comprehensive super-enhancer database of human and mouse. *Nucleic Acids Research*, 51(D1): D280–D290, January 2023. ISSN 1362-4962. doi: 10.1093/nar/gkac968.

Confidential reviewer copy. This manuscript is under double-blind review by ICML 2026. Unauthorized sharing, redistribution, or disclosure is strictly prohibited.

A. Genomic Annotation Dataset Construction for SAE Feature Evaluation

We constructed a comprehensive genomic annotation dataset to evaluate the biological interpretability of sparse autoencoder (SAE) features learned from Evo2 embeddings. All annotations were mapped to the human reference genome GRCh38 (Ensembl release 109).

A.1. Gene Structure Annotations

We extracted gene structure elements from two complementary sources: GENCODE v49 basic annotation (comprehensive gene set) and MANE v1.4 (Matched Annotation from NCBI and EMBL-EBI; clinically relevant transcript set). From the GENCODE GFF3 file (https://ftp.ebi.ac.uk/pub/databases/gencode/Gencode_human/release_49/), we extracted coordinates for transcription start sites (TSS), promoter regions (2kb windows centered on TSS), exons, introns, 5' UTRs, 3' UTRs, coding sequences (CDS), start codons, and stop codons. Intron coordinates were derived by subtracting exon regions from transcript boundaries. Additionally, we incorporated polyadenylation features (polyA sites, polyA signals, and pseudo-polyA) from the GENCODE polyA annotation.

For the MANE dataset, we extracted the same structural elements from MANE.GRCh38.v1.4.ensembl_genomic.gff (https://ftp.ncbi.nlm.nih.gov/refseq/MANE/MANE_human/current/), which represents a minimal set of representative transcripts per protein-coding locus agreed upon by NCBI RefSeq and Ensembl/GENCODE.

A.2. Gene Type Classifications

We extracted 25 distinct gene biotypes based on the "gene_type" attribute in GENCODE annotations. Protein-coding and non-coding RNA classes included:

- Long non-coding RNAs: lncRNA
- Small regulatory RNAs: miRNA, snoRNA, snRNA, scaRNA
- Translation machinery: rRNA, tRNA (from dedicated GENCODE tRNA annotation: gencode.v49.tRNAs.gff3.gz)
- Immunoglobulin genes: IG_C_gene, IG_D_gene, IG_J_gene, IG_V_gene
- T-cell receptor genes: TR_C_gene, TR_D_gene, TR_J_gene, TR_V_gene
- Other functional classes: ribozyme, misc_RNA, TEC (To be Experimentally Confirmed)

PIWI-interacting RNAs (piRNAs) were sourced from piRBase v3.0 (http://bigdata.ibp.ac.cn/piRBase/ (Wang et al., 2019), which provides experimentally validated and computationally predicted piRNA annotations. piRNAs are small non-coding RNAs that play crucial roles in transposon silencing, particularly in germline cells.

A.3. Regulatory Elements

Candidate cis-Regulatory Elements (cCREs): We obtained annotations from the ENCODE Registry V4 (https://screen.wenglab.org/downloads), which integrates DNase-seq, H3K4me3, H3K27ac, and CTCF ChIP-seq data across hundreds of biosamples. The registry classifies cCREs into:

- Promoter-like signatures (PLS): High DNase and H3K4me3 signals, within 200bp of TSS
- Enhancer-like signature (ELS): High DNase and H3K27ac signals, with low H3K4me3 levels if they are located within 200bp of TSS
  - Proximal enhancer-like signatures (pELS): Proximal subset of ELS, within 2kb of TSS
  - Distal enhancer-like signatures (dELS): Distal subset of ELS, >2 kb from TSS
- Chromatin accessible regions with specific features: CA-CTCF, CA-H3K4me3, CA-TF
- Chromatin accessible only (CA) and transcription factor binding only (TF)

Confidential reviewer copy. This manuscript is under double-blind review by ICML 2026. Unauthorized sharing, redistribution, or disclosure is strictly prohibited.

A.4. Enhancer Elements

Super-enhancers, super-enhancer constituent elements, and typical enhancers were obtained from SEdb (Super-Enhancer Database; http://www.licpathway.net:8081/sedb/). Super-enhancers are clusters of enhancers with exceptionally high levels of transcription factor binding and histone modifications.

A.5. CpG Islands

CpG island coordinates were extracted from the UCSC Genome Browser database (**?**Perez et al., 2025). CpG islands are regions with elevated CG dinucleotide frequency often associated with gene promoters.

A.6. CRISPR Targets

Genome-wide CRISPR target sites were included from UCSC Genome Browser annotations (https://hgdownload.soe.ucsc.edu/gbdb/hg38/crispr/).

A.7. Repetitive Elements

We systematically extracted all major repeat element classes from RepeatMasker annotations (https://www.repeatmasker.org/ (Tarailo-Graovac & Chen, 2009), which identifies and classifies repetitive DNA sequences. Categories included:

- Interspersed repeats: SINEs (Short Interspersed Nuclear Elements), LINEs (Long Interspersed Nuclear Elements), LTR retrotransposons, DNA transposons, retroposons, RC (Rolling Circle) elements
- Tandem repeats: Simple repeats, satellites
- Other sequences: Low complexity regions
- RNA-derived repeats: RNA, rRNA, tRNA, snRNA, srpRNA, scRNA

These annotations capture a substantial fraction of the human genome composed of transposable elements and other repetitive sequences, some of which have been co-opted for regulatory functions, while others contribute to genome instability.

A.8. Evolutionary Conservation

Evolutionarily conserved regions were identified using two complementary approaches based on 100-way vertebrate genome alignments from the UCSC Genome Browser (https://hgdownload.soe.ucsc.edu/goldenPath/hg38/database/):

- PhastCons conserved elements: PhastCons identifies conserved elements using a hidden Markov model that considers both substitution rates and spatial clustering of conserved sites. The resulting annotations represent discrete, highly conserved genomic segments.
- PhyloP conservation scores: PhyloP measures evolutionary conservation at individual nucleotide positions using a likelihood ratio test. Positive scores indicate evolutionary conservation, while negative scores indicate accelerated evolution relative to neutral expectation.

A.9. Data Processing and Quality Control

For each annotation category, we performed a standardized processing pipeline to ensure consistency across disparate data sources:

- Coordinate extraction: Genomic coordinates were extracted and standardized into BED format (chromosome, start, end, strand) from diverse source files.
- Sequence extraction: Corresponding DNA sequences were obtained from the human reference genome using the extracted coordinates.

Confidential reviewer copy. This manuscript is under double-blind review by ICML 2026. Unauthorized sharing, redistribution, or disclosure is strictly prohibited.

- Chromosome filtering: Annotations were restricted to standard chromosomes (chr1-22, X, Y, MT) to exclude unplaced contigs and alternative haplotypes.
- Strand information: Strand orientation was preserved where biologically relevant (e.g., for gene features) and maintained as unstranded for features without inherent directionality (e.g., CpG islands, many regulatory elements).

A.10. Evaluation Test Datasets using Genomic Annotations

Using the pooled genomic annotations, we use annotations for chromosome 22 as a heldout dataset for evaluating the SAEs. We define evaluation task for 55 different concepts. We begin by partitioning the chromosome into fixed-length windows matching the Evo2 context size (10 kb) and assigning each window to one or more genomic concepts based on overlap with annotated domains. Each concept is evaluated in either a directional or orientation-agnostic manner, depending on whether the underlying biology distinguishes between plus and minus strands.

For each concept, we identify all genomic windows that intersect annotated regions and rank them by the number of overlapping domains. For directional annotations, we collect 5 representative windows for both the positive and negative DNA strands and reverse complement the negative strands to get the correct 5' to 3' orientation expected for Evo2. For direction-agnostic annotations, we collect 5 representative windows and use the positive DNA strand as the input sequence. To avoid degenerate positives, we exclude windows in which the annotation occupies more than 70% of the context, ensuring that each window has a mix of both positive and negative annotations. Doing this yields test sets with consistent context size and a balanced representation of genomic events.

B. Evaluation Metric Details

B.1. F1 Score using Base-Pair Precision and Domain Recall

For each feature f, we binarize *feature-on* and *feature-off* labels by applying activation threshold $\tau = 0$.

$$\hat{f}(s_i) = \begin{cases} 1, & \text{if } f(s_i) > \tau, \\ 0, & \text{otherwise.} \end{cases}$$

Let $a_i \in \{0, 1\}$ denote the ground-truth annotation label at position i. Precision is evaluated at the base-pair level, and recall is evaluated at the domain level, where a domain d is a contiguous genomic segment annotated as positive for the concept.

$$\text{Precision} = \frac{\sum_i [\hat{f}(s_i) = 1 \wedge a_i = 1]}{\sum_i [\hat{f}(s_i) = 1]}$$

$$\text{Recall} = \frac{|\{d \in D_a : \exists\, i \in d \text{ s.t } \hat{f}(s_i) = 1 \wedge a_i = 1\}|}{|D|}$$

$$\text{F1} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}.$$

For reporting an interpretability score for an annotation using this metric, we select the highest F1 score achieved by any single feature. One of the advantages of this metric is that by measuring recall at the domain level, it accommodates sparse or partial activations within annotated regions, which is well matched to SAE-derived features that may respond to localized motifs rather than entire domains. A feature is therefore credited for detecting a domain even if it activates on only a subset of its positions. However, the metric relies on binarizing feature activations using a fixed threshold, making it sensitive to threshold choice and discarding activation magnitude information. Additionally, in genomic sequences with fixed-length sequence windows, annotated domains may be truncated or only partially observed, which can slightly distort domain-level recall near sequences' boundaries.

B.2. Sparse Probing

Probing is a technique to evaluate how well features can be linearly modified to match a specific task's output (Gurnee et al., 2023). As presented by Gurnee et al., probes utilize basic logistic regression models trained on a small subset of k neurons to identify a particular feature or concept. For each ground-truth annotation a, we define the task of the probe as a

Confidential reviewer copy. This manuscript is under double-blind review by ICML 2026. Unauthorized sharing, redistribution, or disclosure is strictly prohibited.

positive-domain vs. negative-domain classification problem. To reduce the probe to a sparse subset of candidate features, we rank all features $f \in \mathcal{F}$ by the absolute difference in mean activation between positive and negative positions:

$$\Delta_f = \left|\mathbb{E}[f(s_i) \mid a_i = 1] - \mathbb{E}[f(s_i) \mid a_i = 0]\right|.$$

We select the top $k = 10$ features with the largest Δ_f values as probe candidates.

For each selected feature f, we train a one-dimensional logistic regression probe

$$\text{Probe}_{a,f}(s_i) = \sigma(w_f f(s_i) + b_f),$$

where $\sigma(\cdot)$ denotes the sigmoid function. After training the probe, we calculate the area under the precision-recall curve (AUPRC) for the highest performing probe and corresponding feature.

$$\text{Score}_a = \max_{f \in \mathcal{F}_K} \text{AUPRC}(\text{Probe}_{a,f}).$$

Because the probe is univariate, the AUPRC of the probe predictions against the binary annotation labels is equivalent to AUPRC of the raw feature values against the labels. The probe's benefit is being able to represent a feature signal's value as a measure of probability for a particular label. Note that our formulation is slightly different from the one presented by Gurnee et al.; while the authors train a single probe on a combination of the top k features, we train probes on only one feature at a time. Our justification for this choice is motivated by the idealized goal of SAE interpretability: well-trained SAEs should produce monosemantic latents with a near one-to-one correspondence between learned features and underlying genomic concepts. However, it is worth noting that SAEs often exhibit feature splitting, where a high-level concept is divided into multiple more specific latents, or feature absorption, where a parent feature fails to activate and its signal is instead encoded within child features to increase sparsity (Chanin et al., 2024). Unlike the F1 score, sparse probing is advantageous in that it directly utilizes raw feature activation magnitudes to assess linear predictability of annotations, but it is limited in that it only favors features with consistent domain-wide activation and also depends on the top-k selection based on maximum mean difference, which may miss informative features.

B.3. Sparse Event Activation

Some genomic events, such as transcription start sites (TSS) or guanine caps, occur at single base pairs or very short segments at only a handful of locations in the genome. The aforementioned metrics would struggle to reveal associated features as a result of a significant class imbalance due to the vast majority of nucleotides in testing sequences being negative samples. To recover such features, we devise another metric that relaxes the assumption that a feature must occur exactly at an event to it occuring within a specified window W of the ground-truth. We do so by applying a Gaussian kernel weight across activations for each event and finding the mean weighted activation over all events.

$$w(\Delta) = \frac{\exp\left(-\frac{\Delta^2}{2\sigma^2}\right)}{\sum_{\Delta=-W}^{W} \exp\left(-\frac{\Delta^2}{2\sigma^2}\right)}$$

$$A_i = \sum_{\Delta=-W}^{W} w(\Delta)\, f(s_i + \Delta)$$

$$\mu^{(+)} = \frac{1}{|D|} \sum_{i \in D} A_i$$

We then divide this value by the average activation in all remaining areas of the testing sequence (defined as the background).

$$B = \{s_b \,:\, |s_b - s_i| > W \quad \forall i \in D\}$$

$$\mu^{(-)} = \frac{1}{|B|} \sum_{b \in B} f(b)$$

Finally, we compute an enrichment score by taking the log ratio of the two values.

$$S_{\text{ratio}} = \log\left(\frac{\mu^{(+)}}{\mu^{(-)}}\right)$$

Confidential reviewer copy. This manuscript is under double-blind review by ICML 2026. Unauthorized sharing, redistribution, or disclosure is strictly prohibited.

For our results, we use $W = 10$ and $\sigma = 5$.

We additionally evaluate top-ranked features using a windowed F1 score. For each annotated sparse event, we compute a Gaussian-weighted activation within a window of radius W. If this activation exceeds a small threshold ε, the event is considered detected. Event windows are treated as positive instances, yielding counts of true positives (TP) and false negatives (FN). To estimate false positives (FP), we evaluate the same windowed activations at background positions that are at least W bases away from any annotated event. Background windows whose activation exceeds ε are counted as false positives. We then compute precision, recall, and the F1 score in the standard manner. This metric measures whether a feature consistently activates near true events while remaining inactive in background regions, complementing the enrichment score by capturing event-level detection reliability.

The sparse event activation metric is helpful in understanding how well rare annotations are represented in SAE-derived features, but it can be sensitive to the choice of window size W and kernel width σ, and can overestimate feature importance if nearby regions are spuriously activated.

C. Details on Tanimoto/Node2Vec Feature Representation Algorithm

This section describes implementation-specific details for the proposed feature atlas representation method, corresponding to Algorithm 1 in the main text.

Representative Sequence. We create a representative sequence for our algorithm by compiling 500kb of DNA consisting of randomly selected 10kb chunks across chromosome 1 of the human genome. We found that 500kb was a sufficient length to allow the majority of features in the SAE to activate at some point in the sequence, as shown in Figure 8.

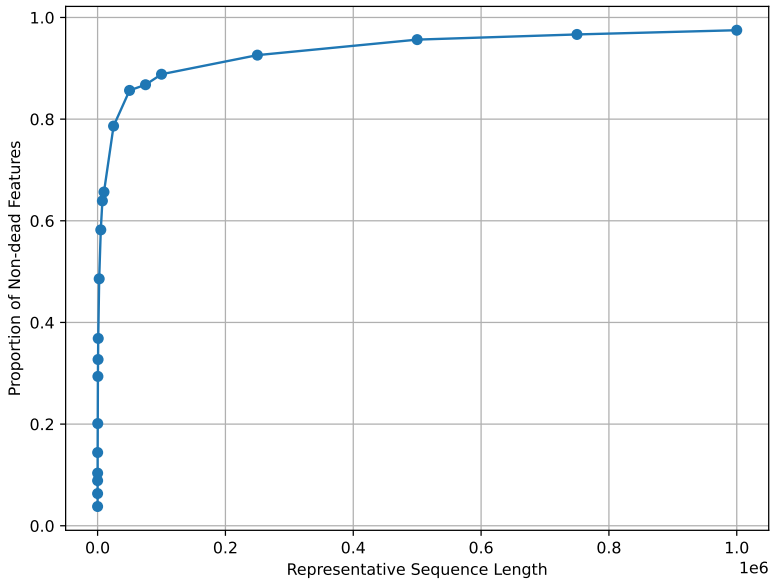

Figure 8. **Proportion of activated features in an SAE across different sequence lengths.** A feature is defined as non-dead if it has a non-zero value at least once within the representative sequence.

Dead Feature Identification. Given feature activation signals $\mathbf{X} \in \mathbb{R}^{F \times L}$, we first identify *dead features*, defined as rows whose activations are identically zero across the entire representative sequence. These features are excluded from similarity computation to avoid numerical instability and reduce computational cost.

Signal Smoothing and Rectification. Activation signals for live features are smoothed along the genomic axis using a wide 1D uniform filter (window size 1000 bp in our experiments), which approximates large-sigma Gaussian smoothing.

Confidential reviewer copy. This manuscript is under double-blind review by ICML 2026. Unauthorized sharing, redistribution, or disclosure is strictly prohibited.

This emphasizes spatially proximal activation while suppressing base-pair–level noise. After smoothing, values are clipped to be non-negative to ensure compatibility with similarity measures designed for sparse, non-negative vectors.

Downsampling for Efficient Neighbor Search. To reduce dimensionality, smoothed signals are downsampled by summing over non-overlapping windows of fixed size (downsampling factor $d = 100$). This produces reduced signals

$$\mathbf{X}_{\text{ds}} \in \mathbb{R}^{F_{\text{live}} \times (L/d)},$$

which preserve large-scale activation structure while enabling a faster approximate nearest neighbor (ANN) search.

Approximate Nearest Neighbor Graph Construction. We use Annoy with angular distance to identify the $k = 80$ nearest neighbors of each live feature in the downsampled space. This step limits Tanimoto coefficient computation to a sparse candidate set of feature pairs and avoids quadratic scaling in the number of features.

Continuous Tanimoto Affinity. For each feature i and its candidate neighbors j, we compute a continuous Tanimoto coefficient

$$K_{ij} = \frac{\mathbf{x}_i \mathbf{x}_j}{\|\mathbf{x}_i\|^2 + \|\mathbf{x}_j\|^2 - \mathbf{x}_i \mathbf{x}_j + \epsilon},$$

where $\epsilon = 10^{-12}$ is a small constant for numerical stability. The resulting values define weighted edges in a sparse affinity matrix. Self-similarities are set to zero.

Affinity Normalization. The affinity matrix is symmetrized and normalized using symmetric degree normalization,

$$\mathbf{K}_{\text{norm}} = \mathbf{D}^{-1/2} \mathbf{K} \mathbf{D}^{-1/2},$$

where $\mathbf{D}$ is the diagonal degree matrix. This normalization stabilizes downstream graph embeddings by preventing high-degree nodes from dominating the representation.

Node2Vec Graph Embedding and Visualization. The normalized affinity matrix is interpreted as a weighted undirected graph, with nodes corresponding to SAE-derived features. We apply node2vec to learn low-dimensional node embeddings via biased random walks (Grover & Leskovec, 2016). We use the node2vec hyperparameters given in Table 3. These

Hyperparameter	Value
Embedding dimension	8
Walk length	60
Number of walks	10
Window	10
Seed	42

Table 3. **Node2Vec Hyperparameters.** We used default settings for other hyperparameters.

embeddings are subsequently reduced to two dimensions using UMAP to obtain the final feature atlas visualization.

Re-integration of Dead Features. Dead features are assigned a common embedding equal to the mean of the live-feature UMAP coordinates. This reflects their lack of informative signal while preserving the original feature indexing in the final embedding $\mathbf{Z} \in \mathbb{R}^{F \times 2}$.

Computational Complexity. Let F_{live} denote the number of live features, L the original signal length, d the downsampling factor, and k the number of nearest neighbors. Signal smoothing and downsampling scale as $\mathcal{O}(F_{\text{live}} \cdot L)$. ANN construction and querying scale approximately as $\mathcal{O}(F_{\text{live}} \log F_{\text{live}})$, while Tanimoto computation on the sparse graph scales as $\mathcal{O}(F_{\text{live}} \cdot k \cdot L/d)$. Node2vec scales linearly in the number of edges and random walks, making the overall method tractable for large SAE dictionaries when $k \ll F_{\text{live}}$.

Confidential reviewer copy. This manuscript is under double-blind review by ICML 2026. Unauthorized sharing, redistribution, or disclosure is strictly prohibited.

D. SAE Training Details

We adopted the BatchTopK sparse autoencoder architecture (Bussmann et al., 2024). The model was implemented using the PyTorch package (Paszke et al., 2019). We use the implementation given by the original authors' open source GitHub repository (https://github.com/bartbussmann/BatchTopK). The weights of the decoder are initialized as the transpose of the encoder, however they are not tied during the training process. SAEs were trained for 5 epochs and optimized using the AdamW optimizer. Unless otherwise specified, the default hyperparameters are specified in Table 4.

Hyperparameter	Value
Activation size	4096
Dictionary size	16384 (x4)
Top-K sparsity	64
Weight decay	1×10^{-3}
Batch size	4096
Number of epochs	5
Learning rate	1×10^{-4}

Table 4. **SAE Hyperparameters.** Default hyperparameter settings for SAE training.

With the exception of the hyperparameter tuning section, all reported SAE results are averaged across 5 random seeds ($seed \in \{3, 42, 123, 545, 898\}$). Apart from the SAE vs. embeddings section, we use layer 19 embeddings from Evo2 to train SAEs.

E. Complete Evaluation Results

Included in Table 5 are the complete set of genomic concepts with annotations available, whether SAEs or embeddings had a better probe AUPRC, and details about the best score (AUPRC) for embeddings vs. SAEs. Sequences were generated with a 10kb context length. Performance by layer for each concept for SAEs are shown in Figure 9 .

Table 5. **Annotation Interpretability with SAEs vs. Embeddings**. Score is defined as the AUPRC of the univariate probe.

Annotation	Winner	Best Score	Best Score Embeddings	Best Score SAE	Best Layer Embeddings	Best Layer SAE
cCREs CA agnostic	Embeddings	0.383814	0.383814	0.353095	1	12
cCREs CA-CTCF agnostic	SAE	0.224629	0.204934	0.224629	1	26
cCREs CA-H3K4me3 agnostic	Embeddings	0.378187	0.378187	0.347278	19	12
cCREs CA-TF agnostic	SAE	0.176624	0.172610	0.176624	19	19
cCREs dELS agnostic	Embeddings	0.742420	0.742420	0.716800	1	12
cCREs pELS agnostic	Embeddings	0.611185	0.611185	0.593239	26	12
cCREs PLS agnostic	SAE	0.354139	0.285848	0.354139	1	19
cCREs TF agnostic	SAE	0.384944	0.366870	0.384944	26	12
cpg islands agnostic	Embeddings	0.700240	0.700240	0.443744	1	26
crispr targets	Embeddings	0.677608	0.677608	0.653877	5	12
IG C gene	SAE	0.590107	0.511927	0.590107	26	26
IG J gene	SAE	0.352377	0.102495	0.352377	30	19
IG V gene	SAE	0.643930	0.546978	0.643930	30	26
mane 3-UTRs	SAE	0.425797	0.352522	0.425797	30	26
mane 5-UTRs	SAE	0.190123	0.067705	0.190123	26	26
mane CDS	Embeddings	0.760606	0.760606	0.590630	30	12
mane exons	Embeddings	0.714247	0.714247	0.580885	30	12
mane introns	SAE	0.776283	0.664571	0.776283	30	26
mane lncRNA genes	SAE	0.740047	0.675204	0.740047	26	26
mane promoters	SAE	0.472369	0.471124	0.472369	30	26

Continued on next page

Confidential reviewer copy. This manuscript is under double-blind review by ICML 2026. Unauthorized sharing, redistribution, or disclosure is strictly prohibited.

Table 5 – continued from previous page

Annotation	Winner	Best Score	Best Score Embeddings	Best Score SAE	Best Layer Embeddings	Best Layer SAE
mane protein coding genes	SAE	0.570290	0.515916	0.570290	1	26
mane snRNA genes	SAE	0.574414	0.132304	0.574414	1	12
mane start codons	SAE	0.076418	0.064044	0.076418	5	5
mane stop codons	SAE	0.041378	0.035318	0.041378	30	12
mane TSS	SAE	0.060206	0.002381	0.060206	19	19
miRNA	SAE	0.089417	0.025104	0.089417	5	19
misc RNA	SAE	0.290934	0.139283	0.290934	19	12
phastcons conserved elements agnostic	Embeddings	0.728946	0.728946	0.628329	30	12
phylop conserved elements agnostic	SAE	0.690984	0.582204	0.690984	1	30
piRNA	SAE	0.321553	0.318972	0.321553	26	26
polyA signal	SAE	0.072608	0.014471	0.072608	19	19
polyA site	SAE	0.078048	0.018911	0.078048	19	19
pseudo polyA	SAE	0.073468	0.010980	0.073468	19	19
repeats DNA	SAE	0.641399	0.497797	0.641399	19	12
repeats LINE	SAE	0.695453	0.575085	0.695453	1	12
repeats Low complexity	Embeddings	0.747392	0.747392	0.689936	5	19
repeats LTR	SAE	0.565804	0.407159	0.565804	26	12
repeats RC	Embeddings	0.150910	0.150910	0.111995	30	30
repeats Retroposon	SAE	0.924757	0.795544	0.924757	26	26
repeats RNA	SAE	0.842460	0.478343	0.842460	19	19
repeats rRNA	SAE	0.561220	0.041521	0.561220	26	19
repeats Satellite	Embeddings	0.431471	0.431471	0.416521	5	12
repeats scRNA	SAE	0.444594	0.075241	0.444594	19	19
repeats Simple	SAE	0.565577	0.486955	0.565577	26	19
repeats SINE	SAE	0.770450	0.542358	0.770450	5	12
repeats snRNA	SAE	0.694804	0.058532	0.694804	26	19
repeats srpRNA	SAE	0.590889	0.313102	0.590889	19	19
repeats tRNA	SAE	0.121381	0.015142	0.121381	5	19
rRNA	SAE	0.895857	0.550721	0.895857	19	19
scaRNA	SAE	0.135387	0.045401	0.135387	1	26
selenocysteine	SAE	0.133637	0.058007	0.133637	1	26
super enhancer elements agnostic	Embeddings	0.532415	0.532415	0.513504	5	12
super enhancers agnostic	SAE	0.598525	0.595709	0.598525	26	19
tRNA	SAE	0.582089	0.079181	0.582089	30	19
typical enhancers agnostic	SAE	0.536285	0.503777	0.536285	12	30

F. Nucleotide Transformer Evaluation Results

We evaluate Nucleotide Transformer v3's 8M parameter model available in HuggingFace (https://huggingface.co/InstaDeepAI/NTv3_8M_pre). The model was evaluated with SAEs at layers 4, 9, and 13. Layer 4 corresponds to an intermediate Down Res block, layer 9 corresponds to the embedding after the transformer layer, and layer 13 corresponds to an intermediate Up Res block. The embedding dimension is 256 across all layers. SAEs for Nucleotide Transformer were trained for 10 epochs, with a batch size of 1024 for layers 4 and 13 and a batch size of 64 for layer 9. Because the Nucleotide Transformer operates on strided convolutional windows rather than individual base pairs at the intermediate layers, we aggregated base-pair resolution annotations into window-level labels using max pooling to match the model's effective resolution when evaluating SAE features. SAE features vs. embeddings comparisons across concepts are shown in Figure 11.

Confidential reviewer copy. This manuscript is under double-blind review by ICML 2026. Unauthorized sharing, redistribution, or disclosure is strictly prohibited.

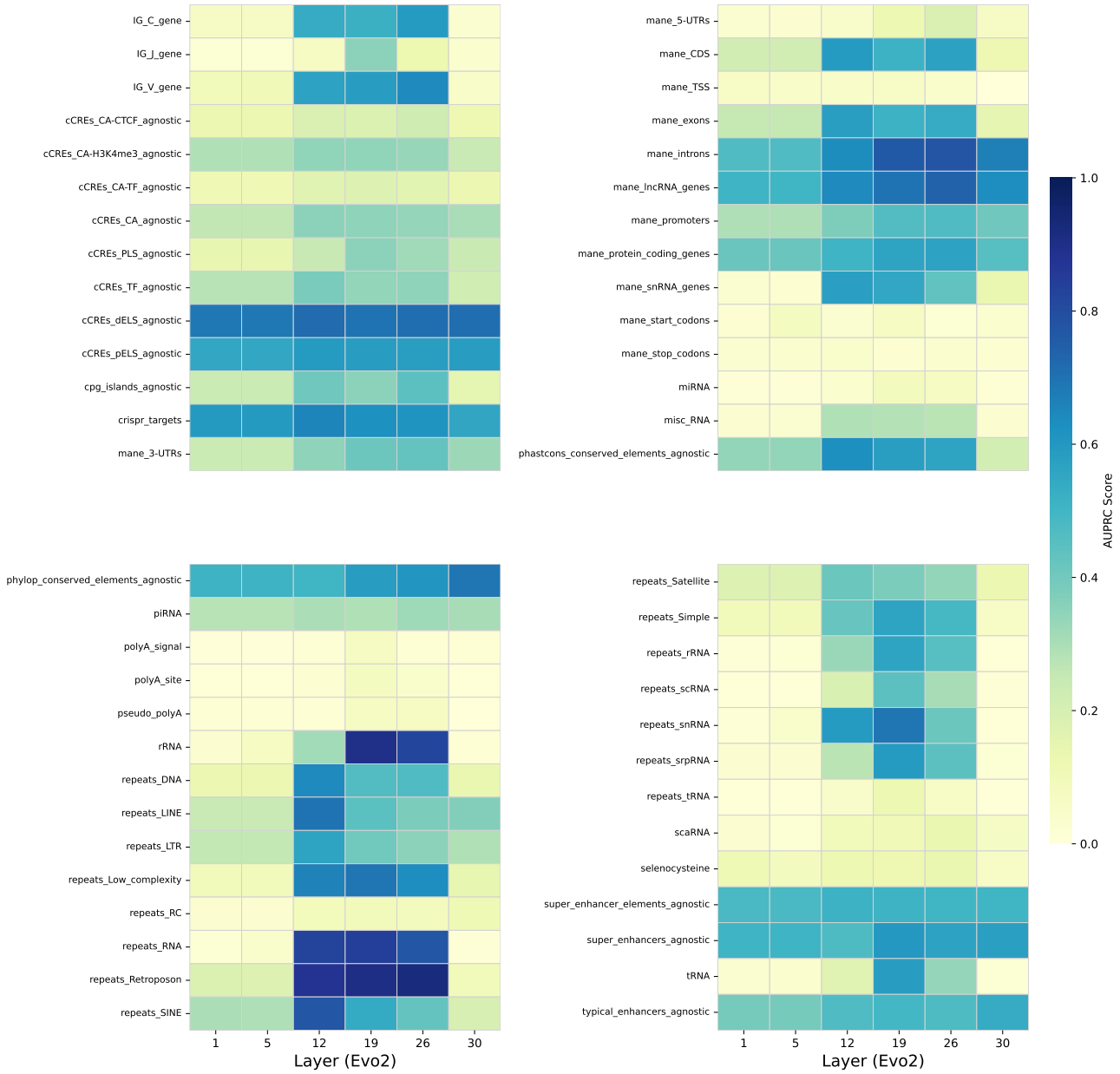

Figure 9. **AUPRC score of SAEs across different layers of Evo2.**

G. Sparse Event Evaluation Results

In Table 6, we show the performance of some annotations that are considered sparse or occur very rarely. We use the sparse event activation metric for evaluation.

H. Training Genomes

H.1. Sequences from Reference Genomes

We used reference genomes available on NCBI for four different organisms spanning different evolutionary distances to humans. From most distant to most proximal:

- E. Coli Genome - https://www.ncbi.nlm.nih.gov/datasets/genome/GCF_000005845.2/

Confidential reviewer copy. This manuscript is under double-blind review by ICML 2026. Unauthorized sharing, redistribution, or disclosure is strictly prohibited.

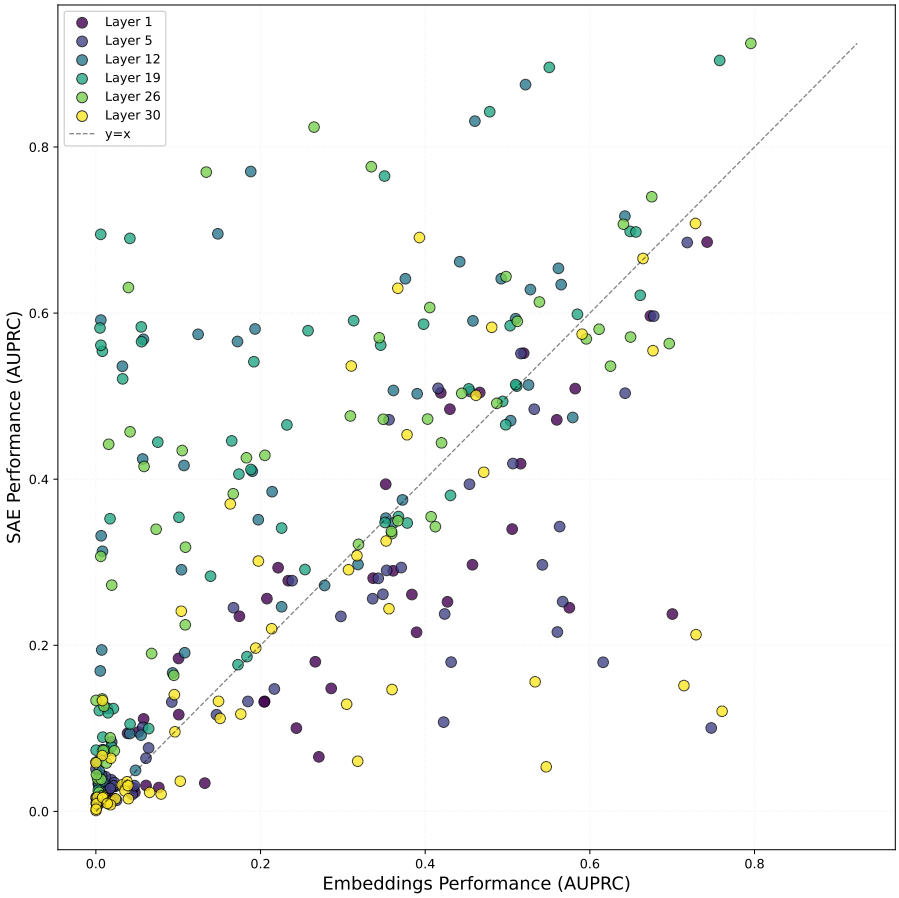

Figure 10. **Feature vs. embeddings using all evaluations (Evo2).** Each point in the scatterplot represents an evaluated genomic annotation at a given layer. Its coordinates represent the sparse probing AUPRC metric using SAE-derived features and the embeddings.

	1	5	12	19	26	30
mane TSS	4.270 (0.01)	5.321 (0.07)	6.331 (0.06)	8.374 (0.08)	3.758 (0.01)	6.904 (0.01)
mane start codons	4.341 (0.02)	6.418 (0.25)	6.820 (0.09)	8.408 (0.13)	3.241 (0.02)	7.142 (0.03)
mane stop codons	4.592 (0.02)	6.302 (0.14)	7.986 (0.15)	8.784 (0.12)	4.154 (0.01)	20.506 (0.05)
miRNA	3.751 (0.17)	5.233 (0.22)	6.340 (0.11)	9.575 (0.33)	5.829 (0.19)	18.911 (0.13)
selenocysteine	6.281 (0.47)	7.519 (0.41)	9.020 (0.33)	10.153 (0.34)	9.546 (0.11)	15.221 (0.13)

Table 6. **Sparse event activation metric for sparse genomic concepts.** Performance of SAEs were evaluated using Evo2 embeddings at 6 different layers. Top features were selected using highest F1; while we report the enrichment score values, we include the corresponding F1 score in parentheses.

- C. Elegans Genome - https://www.ncbi.nlm.nih.gov/datasets/genome/GCF_000002985.6/
- Mouse Genome - https://www.ncbi.nlm.nih.gov/datasets/genome/GCF_000001635.27/
- Human Genome - https://www.ncbi.nlm.nih.gov/datasets/genome/GCF_009914755.1/

To generate training sequences for the SAEs, we randomly sample 500 sequences each of length 10kb from each genome. For the human genome, to prevent any data leakage, we omit chromosome 22 as it is used for evaluation. Every other sequence is reverse-complemented. These sequences are passed into Evo2 and embeddings are extracted at layer 19.

Confidential reviewer copy. This manuscript is under double-blind review by ICML 2026. Unauthorized sharing, redistribution, or disclosure is strictly prohibited.

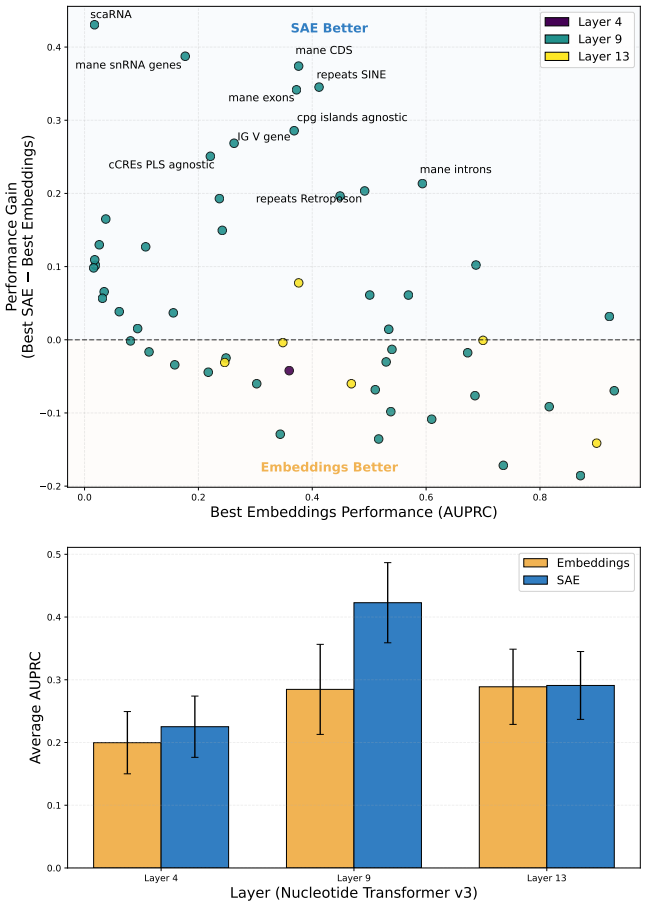

Figure 11. **Comparison of SAE features and Nucleotide Transformer (NT) embeddings aligning with genomic annotations.** *(Top)* Each point corresponds to an annotation and is colored by the NT layer that yielded the highest performance for that concept. Gain is the difference between the best SAE probe AUPRC and the best embeddings probe AUPRC (across all three evaluated layers). *(Bottom)* Average performance of embeddings and SAEs of all benchmarked concepts at each layer, with standard deviation across concepts.

H.2. Evaluation Genes

For evaluation, we use the genes given in Table 7. These genes include a mix of highly conserved and weakly conserved genes. For each gene, we select a 10kb chunk that contains a sufficient number of exon-intron junctions.

H.3. Best Performance of Genes

Using the probe metric, we determine which training genome yielded the best performance for each individual gene. The distribution of best training genomes is highlighted in Figure 12 . We observe that most genes (10/17) have a peak performance on the mixed-species trained SAE. Interestingly, when using the F1 score metric instead, there is a more even distribution of genes winning across the mammal-trained SAEs (Figure 13). The probe favors features that have full domain activation whereas the F1 favors features that might have partial domain coverage, such as activating on a few relevant motifs. This discrepancy may illustrate how the top-performing features for each of the SAEs would respond to exons, with localized motif-targeting features emerging with SAEs trained on mammals and more robust full exon-activating features emerging with SAEs trained on human DNA.

I. Context Length

To assess context length influences on SAE performance, we start from the human sequences used for training SAEs from the training genomes experiment (Appendix H), containing 500 genomic sequences, each 10 kb in length. To control dataset size, only the first 100 sequences are retained, corresponding to 1 Mb of total DNA. Each retained sequence is then

Confidential reviewer copy. This manuscript is under double-blind review by ICML 2026. Unauthorized sharing, redistribution, or disclosure is strictly prohibited.

Gene ID	Gene Name	Gene Start	Gene End	Chunk Start	Chunk End	PhastCons Score
2033	EP300	41092591	41180076	41159565	41169564	0.950311
613	BCR	23180508	23318036	23279565	23289564	0.915888
2130	EWSR1	29268267	29300520	29289565	29299564	0.904830
6721	SREBF2	41833104	41907304	41869565	41879564	0.878963
4174	MCM5	35400139	35455030	35409565	35419564	0.873817
4771	NF2	29603555	29698599	29669565	29679564	0.860870
6523	SLC5A1	32043260	32113028	32079565	32089564	0.751534
5435	POLR2F	37953662	38041914	37949565	37959564	0.654839
9463	PICK1	38057254	38075700	38069565	38079564	0.589985
23765	IL17RA	17084999	17115692	17097067	17107066	0.522173
6948	TCN2	30607173	30627270	30609565	30619564	0.517274
80339	PNPLA3	43923804	43947581	43919565	43929564	0.401349
1312	COMT	19941771	19969974	19959565	19969564	0.277810
11274	USP18	18150169	18177396	18169130	18179129	0.272388
158	ADSL	40346499	40387526	40359565	40369564	0.202278
1439	CSF2RB	36913627	36940438	36929565	36939564	0.193015
200316	APOBEC3F	39040863	39055971	39039565	39049564	0.138232

Table 7. **Selected genes for evaluation on chromosome 22.** Genes are sorted by most conserved to least conserved. Best-overlapping chunk coordinates and PhastCons conservation scores are also indicated.

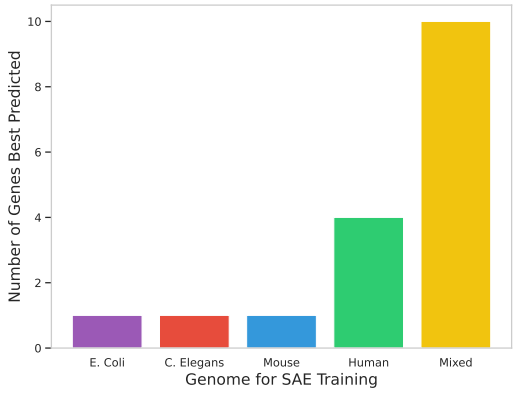

Figure 12. Distribution of best training genome for SAEs based on maximum probe performance (AUPRC) for each gene.

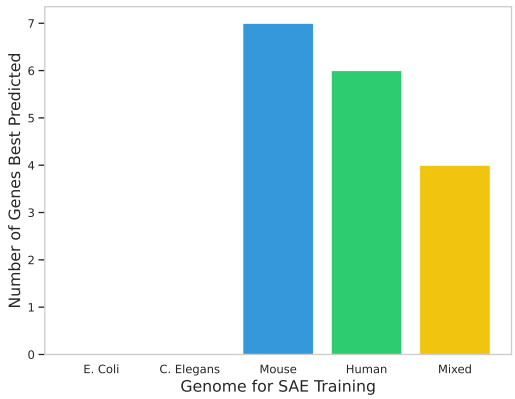

Figure 13. Distribution of best training genome for SAEs based on maximum F1 score for each gene.

subdivided into non-overlapping contiguous chunks at multiple resolutions (50, 100, 250, 500, and 1,000 bp). All chunked datasets are derived from the same underlying 1 Mb of genomic DNA. Changing the chunk size does not introduce new sequence content or remove existing bases; it only alters how the DNA is partitioned.

Confidential reviewer copy. This manuscript is under double-blind review by ICML 2026. Unauthorized sharing, redistribution, or disclosure is strictly prohibited.

Context length can play a critical role in the features that are derived, as embeddings for a given segment of DNA can change based on the window of DNA that was fed into the gLM. As illustrated in Figure 14, embeddings can change in particular stretches of a query sequence, provided with different context sizes. We observe that the first few nucleotides of a sequence (∼500bp) tend to have a higher sensitivity to context length.

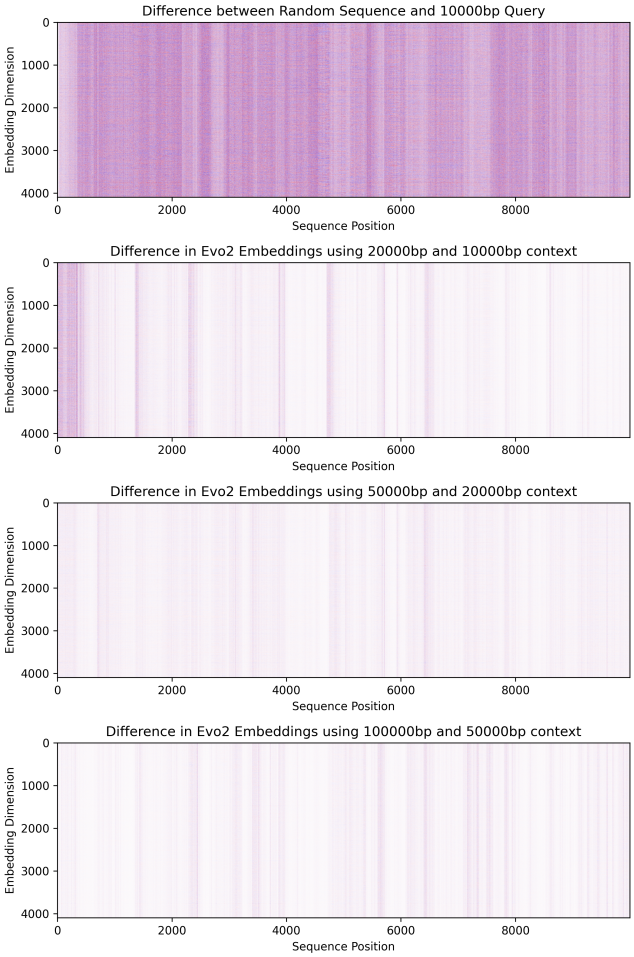

Figure 14. Differences in Evo2 layer 26 embedding values with different context windows.

J. Software and Data

We provide code for the review process at the following Anonymous GitHub repository: https://anonymous.4open.science/r/icml-glm-interpretability/. The repository will be updated with complete scripts and datasets in an accepted version of this paper.

K. Computational Requirements

We use Evo2 7B model to collect embeddings. The model was run on a high performance compute cluster using an Apptainer image of Evo2 (https://zenodo.org/records/15194473). We ran Evo2 on an NVIDIA H100 GPU. SAEs were trained on one GPU, and we parallelized our training across a collection of 2 H100 GPUs.

Confidential reviewer copy. This manuscript is under double-blind review by ICML 2026. Unauthorized sharing, redistribution, or disclosure is strictly prohibited.