SPARSE AUTOENCODERS REVEAL INTERPRETABLE FEATURES IN SINGLE-CELL FOUNDATION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Single-cell foundation models (scFMs) hold promise for applications in cell type annotation and data integration, but their internal mechanisms remain poorly understood. We investigate the structure of these models by training sparse autoencoders (SAEs) on the hidden representations of two widely used scFMs, scGPT and scFoundation. The learned features reveal diverse and complex biological and technical signals, which emerge even in pre-trained models. We also observe that the encoding of this information differs between scFMs with distinct training protocols and architectures. Further, we find that while many features capture the information about cell types across several studies, they often fall short of unifying it into a single generalized representation. Finally, we demonstrate that SAE-derived features are causally related to model behavior and can be intervened upon to reduce unwanted technical effects while steering model outputs to preserve the core biological signal. These findings provide a path toward more interpretable and controllable single-cell foundation models.

1 INTRODUCTION

Single-cell foundation models (scFMs) have emerged as a powerful tool in computational biology to analyze cellular states and behavior (Bunne et al., 2024; Szałata et al., 2024). These models are typically trained on large-scale single-cell RNA sequencing datasets using self-supervised learning objectives (Cui et al., 2024; Hao et al., 2024; Rosen et al., 2023). They can then be applied to several downstream tasks such as cell type annotation (Theodoris et al., 2023; Heimberg et al., 2025), batch integration (He et al., 2025; Cui et al., 2024) or perturbation prediction, including drug response (Hao et al., 2024; Theus et al., 2024), and gene perturbation response (Adduri et al., 2025). The promise of scFMs lies in their ability to capture complex, non-linear relationships in high-dimensional transcriptomics data and generalize across different cell types, tissues, and experimental conditions.

Single-cell foundation models have found growing utility in single-cell analysis, yet many remain difficult to interpret. Like other large models, they often function as black boxes, making it unclear how their predictions are made (Fu et al., 2024). As these models are increasingly explored for generating biological insights, understanding what drives their predictions becomes important. This lack of transparency is particularly problematic given that their architectures are largely inherited from natural language processing models, with only minimal adaptation to the biological context (Theodoris et al., 2023; Yang et al., 2022). Moreover, despite their theoretical promise and computational complexity, several limitations of scFMs have been identified. Recent benchmarking studies have revealed mixed performance, with some finding that linear models can match or outperform scFMs in tasks such as cell type classification (Kedzierska et al., 2023) and perturbation response prediction (Ahlmann-Eltze et al., 2024; Csendes et al., 2024; Kernfeld et al., 2024), though performance appears to be task- and dataset-dependent. Additionally, scFMs often require fine-tuning to achieve practical usability (Liu et al., 2024; Steiner et al., 2025; Ovcharenko et al., 2025). Gaining a deeper understanding of how these models make predictions is essential for guiding future improvements in their design and training strategies.

Recent advances in mechanistic interpretability have introduced sparse autoencoders as a powerful technique for decomposing learned representations into interpretable, sparsely activated features that correspond to meaningful concepts (Cunningham et al., 2023). This approach has yielded insights

into the internal workings of transformer-based language models (Templeton et al., 2024), DNA language models (Guan et al., 2025), and protein language models (Simon & Zou, 2024; Adams et al., 2025). In this work, we utilize sparse autoencoders to understand the internal representations of scFMs, aiming to uncover the biological features these models learn, determine whether their representations align with biological knowledge, and assess the extent to which they encode technical artifacts.

Our main contributions are: 1) We show that pre-trained scFM have a complex and meaningful understanding of cell biology 2) We show how training procedures can affect the encoding of information within the model 3) We propose that pre-trained scFMs struggle to learn unified representations of cell types. Instead, cells from certain studies may not be fully integrated into a single representation, which can limit their generalizability. 4) We demonstrate that SAE-derived features are causally related to model behavior and can be intervened upon to improve batch integration in scFMs.

2 Background

2.1 Sparse autoencoders

Sparse autoencoders (SAEs) are neural networks that learn interpretable representations by constraining most latent neurons to be inactive. They use an encoder-decoder architecture where the encoder maps input data to a sparse latent representation and the decoder reconstructs the original input. The sparsity constraint ensures only a few neurons activate simultaneously, promoting the discovery of monosemantic features that correspond to single, interpretable concepts rather than polysemantic neurons that respond to multiple unrelated patterns.

Recent work has shown SAEs effectively extract interpretable features from complex models. Cunningham et al. (2023) applied SAEs to transformer residual stream activations, revealing sparsely activating features corresponding to interpretable concepts like specific topics and grammatical structures. Recent work from Anthropic (Bricken et al., 2023; Templeton et al., 2024) has further advanced this approach by successfully scaling SAEs to much larger models, including applications to 34-billion-parameter language models. Using improved training techniques, they have extracted millions of interpretable features corresponding to high-level concepts, entities, and reasoning patterns. This work demonstrated that monosemantic features remain interpretable at the scale of state-of-the-art language models, establishing SAEs as a viable tool for understanding large language models.

2.2 Sparse autoencoders for single-cell foundation models

The application of SAEs to biological data has recently emerged, particularly for interpreting single-cell RNA sequencing (scRNA-seq) foundation models. Recent studies have demonstrated that SAEs can successfully decompose cell embeddings from pretrained single-cell models to extract interpretable biological features (Schuster, 2025; Claye et al., 2025). These approaches have shown that SAEs can identify and steer biological processes and signals from the final cell representations. Schuster (2025) introduced automated interpretability tools for linking SAE features to biological concepts using gene sets, while Claye et al. (2025) developed methods to interpret latent concepts by identifying contributing genes through counterfactual perturbations and leveraging textual gene descriptions from ontologies. Both studies focused primarily on analyzing the cell embedding space of pretrained single-cell models.

Our approach extends this work in several key directions. Unlike previous studies that analyze final cell embeddings, we apply SAEs to intermediate token representations during the model's forward pass, capturing richer biological information before it is compressed into cell-level summaries. We conduct comparisons across two foundation models (scFoundation and scGPT under several fine-tuning protocols) and diverse datasets, enabling analysis of how training strategies and data influence learned representations. We also provide a more detailed categorization of discovered features, distinguishing gene- from cell-specific patterns, and introduce novel applications of SAE-based steering to address technical artifacts such as batch effects while preserving biological signals.

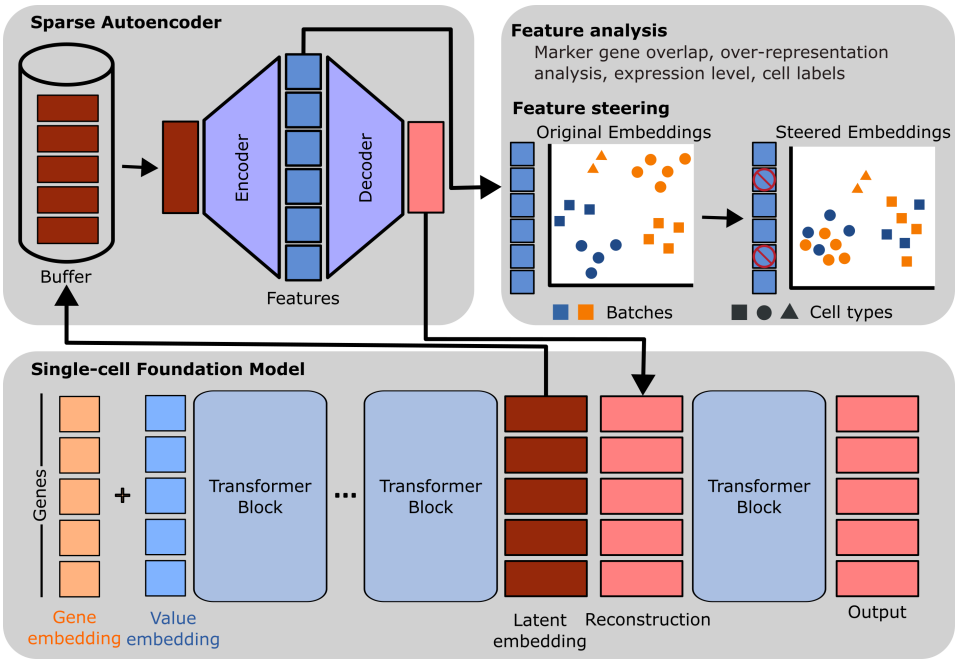

Figure 1: Framework used to train and evaluate sparse autoencoders from scFM activation tokens.

3 METHODS

3.1 ARCHITECTURE AND TRAINING

Our approach involved training sparse autoencoders (SAEs) on intermediate representations from pre-trained single-cell foundation models to extract interpretable features. We performed inference with the foundation model on scRNA-seq datasets and extracted token representations from specific transformer layers. For most experiments we used layer 5 for steering and layer 10 (out of 12 total layers) for feature interpretability analysis. These layers were selected based on systematic layer-wise experiments across all scGPT layers (see Appendix A.5 Figures 10, 12, 13). Layer 5 was chosen as it falls within the middle layers (5-8) that consistently achieved optimal steering performance. Layer 10 was selected for interpretability analysis as later layers exhibit higher SAE reconstruction quality and more structured representations that align with our feature analysis methodology. The residual stream activations were then forwarded to the SAE for training as shown in Figure 1. Following the recommendations of Bricken et al. (2023), we ensured diverse token sampling and large batch sizes. We sampled 250 tokens per cell and aggregated them into a buffer, then constructed training batches of size 8,192 with tokens drawn from varied cellular contexts.

We evaluated the features of two single-cell foundation models: scGPT and scFoundation. For scFoundation, only the pre-trained variant was used since the fine-tuning code was not available. Throughout this paper, scGPT fine-tuning refers to continued training of the pre-trained model on a new dataset using only the self-supervised objectives GEP (Gene Expression Prediction) and GEPC (GEP for Cell Modelling) as described in the original scGPT paper, allowing the model to learn the target dataset's distribution without specific label-based tasks. The DAR (Domain Adaptive Regularization) objective was additionally used when explicitly stated. We used the BatchTopK architecture (Bussmann et al., 2024) for all SAEs, which was chosen because it performs best in language models and showed the same advantage in the single-cell setting in our experiments (see Appendix A.3.1).

The SAEs were trained with Adam (Kingma & Ba, 2014) using hyperparameters adjusted according to dataset size. For smaller datasets (Pancreas, Lung, Immune) we used a learning rate of 0.001, while for larger datasets (COVID-19, CellxGene) we used 0.0001. For most datasets we used a latent dimension of 512 with sparsity of 10, except CellxGene, for which we used 1024 dimensions

and sparsity of 20 to accommodate its greater variability. In general, we kept feature spaces small and sparse, since larger feature spaces tend to split features into overly fine-grained concepts that are difficult to interpret. This effect is more pronounced in smaller datasets, which show limited variability and thus support fewer distinct features. The complete hyperparameter settings for each dataset are provided in Appendix A.3.1.

The scFM training loss was consistently high and did not provide sufficient signal for computing the standard 'loss recovered' SAE evaluation metric (Bricken et al., 2023). To address this limitation, we developed the Embedding Recovery Score, which directly measures the impact of token-level reconstruction quality on downstream task performance through the cell embeddings (see Appendix A.3.2 for definition).

3.2 Datasets

We trained and evaluated sparse autoencoders on five different datasets: CellXGene Census (Program et al., 2025), COVID-19 (Yoshida et al., 2022), Immune, Pancreas, and Lung (Luecken et al., 2022). The COVID-19 and CellXGene Census datasets were mainly used for feature identification, while the Pancreas, Immune, and Lung datasets were used for steering experiments. Further details can be found in Appendix A.1.

3.3 Feature Analysis

We analyzed SAE features mainly through two complementary approaches: cell-level associations and functional enrichment analysis.

Cell-level associations: To evaluate whether features capture cell identity, we examined relationships between feature activations and cell type labels. The SAE produces a continuous activation value for each feature at each gene position within a cell. We aggregated these gene-level activations to the cell level by max-pooling across all genes, yielding a single activation score per feature per cell. To compute concept alignment metrics (adjusted mutual information and F1 scores), we applied activation thresholds to convert these continuous cell-level scores into binary classifications (feature active/inactive for each cell), which were then compared against ground-truth cell-level labels. We examined multiple threshold values, as the interpretation of what constitutes an "activation" can vary, and often focused on strongly activated cells to identify core feature concepts.

Functional enrichment analysis: To interpret the biological relevance of features, we tested whether genes with strong feature activations were enriched for known biological categories. We performed over-representation analysis (ORA) (Subramanian et al., 2005) using Gene Ontology gene sets (Liberzon et al., 2015) to assess enrichment across biological processes, cellular components, and molecular functions. Additionally, we used PanglaoDB marker sets (Franzén et al., 2019) to examine enrichment of cell type-specific expression markers.

Additionally, we used Spearman's rank correlation to assess whether features were associated with high gene expression values, and quantified feature enrichment for specific gene families by calculating what fraction of each feature's activations came from genes in those families.

3.4 Feature Steering

We used feature steering to demonstrate that SAE features can be manipulated to reduce batch effects while preserving the biological signal. Our approach involved identifying batch-correlated features and systematically suppressing their activation during inference. This steering procedure serves as a mechanistic probe to test whether identified features contribute to model behavior: if suppressing a feature systematically alters model outputs in predictable ways, this provides evidence that the feature actively encodes that property rather than representing spurious correlation.

We first identified SAE features that strongly correlate with batch labels by calculating AMI scores as described in the previous section. To perform steering, we passed the dataset through the scFM to obtain intermediate token representations, then projected these into feature space using the SAEs. We clamped the identified "batch features" to zero or a negative value whenever they activated on a gene token. To suppress their contribution while avoiding disproportionate downstream effects from extreme negative values, we set this clamp value to -2 for most experiments. The modified

feature activations were then projected back into token space via the SAE decoder and fed into the remaining scFM layers to generate new cell embeddings.

We evaluated the effectiveness of our steering approach by comparing the resulting embeddings to several baselines: the original scFM embeddings (without intervention), PCA on the raw gene expressions (no batch removal), and scVI. scVI (Lopez et al., 2018) is a widely used conditional variational autoencoder-based method for batch correction in single-cell data and is considered among the leading approaches in this area. We include scVI as a reference point to contextualize the magnitude of batch effects in the data rather than as a competitive baseline. We assessed both batch integration quality and preservation of biological variation using the metrics described in Appendix A.2.

4 RESULTS

4.1 SPARSE AUTOENCODERS FIND INTERPRETABLE CONCEPTS

Sparse autoencoders trained on scGPT (both pre-trained and fine-tuned) and on scFoundation produced similar sets of features. These features can be grouped into two broad categories: **gene-specific** and **cell-specific**. Example features for each category for different models and datasets can be found in Appendix A.4. To quantify the prevalence of different feature types, the first 100 of the 1024 features extracted from the pre-trained scGPT model on the CellxGene Census were manually annotated using the methodology described in section 3.3. Features were categorized based on their cell-level associations with known cell types or batch labels, their functional enrichment for Gene Ontology terms and PanglaoDB marker sets, and their enrichment for specific gene families. This analysis identified 26 features associated with cell types, 2 with batch effects, 29 enriched for biological processes and 7 for curated gene sets. Approximately one third of these features were difficult to interpret, suggesting that they may capture biological variation not represented in gene ontology sets or reflect heuristic patterns that are less directly interpretable. The distribution of our annotated feature categories is shown in Figure 2.

Gene-specific features reflect properties of individual genes, independent of the cells they are expressed in. Some correlated with specific expression levels of individual genes. In scFoundation, these features were more strongly linked to low expression. In scGPT, they appeared across several expression levels, and the effect was stronger, likely because gene expression is represented in bins. This binning ensures that high expression is always mapped to the same bin, a pattern that the scFM learned and used. Other features correlated with individual genes or gene families such as ribosomal, mitochondrial, human leukocyte antigen, or immunoglobulin genes. A third group we identified captured biological processes such as the cell cycle, defense response, and apoptosis.

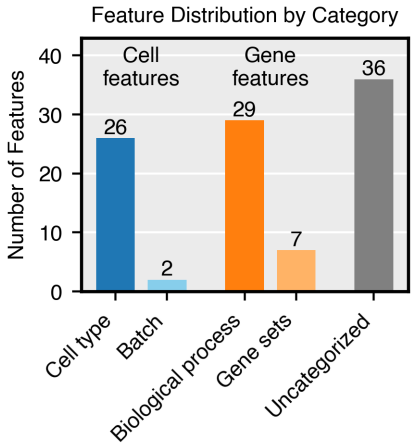

Figure 2: Distribution of categories of 100 random concepts of pre-trained scGPT on the CellxGene dataset

Cell-specific features capture properties of cells rather than individual genes. They mostly arise from contextual information added to gene tokens by the transformer, which is trained to reconstruct cell identity from individual tokens. This information may also be distributed across many tokens rather than being concentrated in the most relevant ones. Our analysis showed features corresponding to many annotated cell types. In the COVID-19 dataset, broader cell types were consistently represented by at least one feature in all scFM models tested, and were often enriched for biological processes and marker genes associated with cell identity. Finer cell type annotations were less consistent. Some appeared only in scGPT, others only in scFoundation, and some not at all. The fine-tuning of scGPT on the COVID-19 dataset did not noticeably improve the F1 or AMI scores of these features. However, the classification score when using the fine-tuned cell embeddings improved noticeably. It is also important to note that internal cell-type representations in the scFM did not always align with our labels.

For example, both models produced features linked to monocytes and dendritic cells together, or only to monocytes, but none exclusively to dendritic cells. While the best-scoring cell type features were similar between scGPT and scFoundation, scGPT showed a larger variety of these features that had different key characteristics. Additional feature categories included disease-related features that activated more frequently in COVID-19 patients and batch-related features that activated on specific studies, donors, or sequencing technologies.

4.2 Concepts from pre-trained models are varied and meaningful

The following findings highlight examples of how pre-trained scFMs capture rich features that are both biologically meaningful and reflect technical effects in sequencing technologies, even though these models do not excel in zero-shot tasks.

Features Capture Diverse Aspects of Cell Identity. Pre-trained scFM models captured multiple distinct aspects of cell identity. For each cell type, we observed several features that were strongly correlated with the cell type label, but these features differed in their characteristics. We use B-cell features from pre-trained scGPT to illustrate the diversity of these feature groups (see Table 1).

Some features activated on many different genes across a wide range of expression levels. These represent general features that are frequently active, largely independent of specific gene identity, and broadly responsive within the cell type. Feature 478 exemplifies this pattern.

Other features targeted a small set of highly expressed genes, often marker genes. These represent highly specific cell type signatures closely tied to canonical marker gene activity. Feature 363 illustrates this pattern: 26 of its 83 genes with activations above 0.5 are known B-cell markers, while 38 others are either immunoglobulin genes or major histocompatibility complex genes, both commonly enriched in B cells. This enrichment is statistically significant (adjusted p-value = 4.94e-20), and gene ontology analysis showed enrichment in B-cell-mediated immunity (adjusted p-value = 2.06e-17).

A third type of feature activated on low-expression gene subsets and may act as a negative signal for other cell types. Feature 151, for example, was moderately enriched in marker genes of T cells, monocytes, macrophages, megakaryocytes, and NK cells but notably not B cells. This pattern suggests that the feature captured the absence of non-B-cell marker genes, effectively serving as a negative signal for other cell identities within B cells.

Some features activated on gene subsets that are not immediately linked to cell identity, but the expression patterns of these gene subsets can still distinguish the corresponding cell type from others. These features often consisted of subgroups of ribosomal genes (e.g., feature 270).

Table 1: Selected features predictive of B cell identity with key characteristics.

Feature	AMI	F1	Expression (mean $\pm$ SD)	Unique genes	Ribosomal (%)
478	0.97	0.99	18.68 ± 6.13	1485	0
363	0.95	0.99	42.61 ± 5.79	83	0
151	0.99	1.00	4.27 ± 2.94	2698	2
270	0.85	0.95	43.92 ± 5.43	187	85

Features Encode Biological Processes from Unseen Conditions. Pre-trained scFM models capture distinct aspects of cellular function and state, including features that reflect specific biological processes or abnormal cell states not present in the training data. Despite being trained only on healthy cells, some features activated in response to disease-associated cellular states.

For example, one feature activated predominantly in monocytes and dendritic cells from patients with post-COVID-19 disorder (see Appendix Figure 6). The gene set that most strongly activated this feature was highly enriched in inflammation-related pathways (adjusted p-value: 1.19e-23). This finding aligns with studies showing that monocytes and dendritic cells in post-COVID patients can remain in a persistently activated inflammatory state even months after the acute infection has resolved.

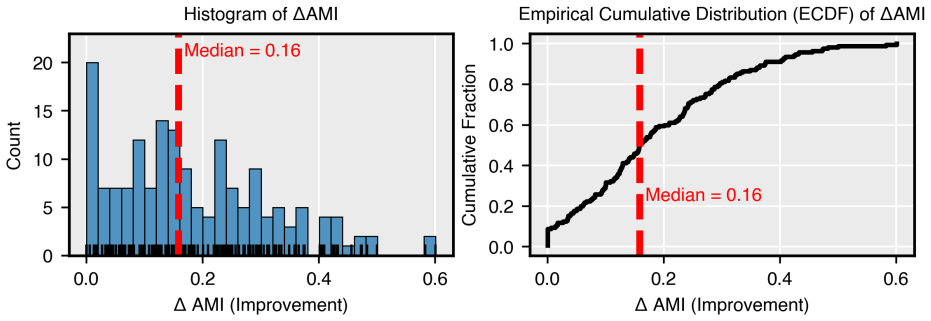

Figure 3: Distribution of the ΔAMI calculated between feature activations and two cell type annotations. Values above zero indicate features that better capture cell types within specific studies than across all studies.

Features Capture Technical Aspects of Sequencing Protocols. Pre-trained scFM models also capture systematic biases inherent in sequencing technologies. These biases manifested as features that correlate with technical variables such as gene length and GC content.

In the Pancreas dataset, we observed that cells processed with the sequencing protocol "SMARTer" showed distinct technical signatures compared to other sequencing methods. Specifically, these cells exhibited a stronger positive correlation between gene count and gene length, and a stronger negative correlation with GC content, compared to the dataset as a whole. The scGPT feature with one of the highest AMI between its activations and the SMARTer sequencing label captured exactly this technical bias: it activated on highly expressed genes with significantly greater length and lower GC content than the dataset average (see Appendix Figure 7).

4.3 Cell type features show variable generalization across datasets

The CellXGene Census dataset aggregates many studies using different experimental protocols and sequencing technologies. A key requirement for single-cell foundation models is their ability to generalize beyond protocol-specific technical artifacts and capture shared biological principles. While technical effects persist in learned representations unless explicitly addressed during training, robust models should identify biological concepts that extend across these experimental variations.

Our analysis revealed that features learned from the census dataset showed varying degrees of generalization across different studies and sequencing platforms. Although some features demonstrated consistent activation patterns across the entire data set, many others appeared to be more strongly associated with specific subsets of the data. Specifically, SAE-derived cell type concepts showed variable activation patterns, with some activating on cells from particular studies while showing reduced activity on the same cell types from other studies.

To quantify this observation, we compared the AMI between feature activations and complete cell type annotations with the AMI between feature activations and cell type subsets spanning a subgroup studies. In most cases, these cross-study subsets showed stronger alignment with feature activations than the complete cell type categories (see Figure 3). This pattern suggests that while the learned representations can capture cellular identity across multiple studies, they may not always consolidate these signals into unified concepts that cover all instances of a given cell type across the entire dataset.

4.4 Cell embeddings can be integrated by deactivating specific features

Table 2 and Appendix Tables 16 and 17 report benchmarking results of all approaches on batch integration quality and preservation of biological variation across the three datasets. For the steering experiments reported in these tables, we clamped the 50 features with highest AMI score with respect to batch labels for fine-tuned models, but only the top 20 features for pre-trained models. We used fewer features for pre-trained models because there the concepts are less well represented.

As expected, unintegrated PCA embeddings had the lowest performance, while the explicit batch integration method scVI achieved the highest scores.

Steering improved batch correction for fine-tuned models across all datasets without substantial loss in biological conservation. On the pancreas dataset, steering the fine-tuned scGPT outperformed both the native DAR batch correction of scGPT and the unsteered fine-tuned model, with improved biological conservation. Figure 8 shows Uniform Manifold Approximation and Projections (UMAPs) (Healy & McInnes, 2024) of embeddings from unaltered and steered fine-tuned scGPT, colored by cell type and sequencing technology. These plots show how steering reduces batch effects and improves clustering by cell type. Pre-trained steering was only consistently successful on the pancreas dataset. This likely reflects the stronger batch effects in this dataset and the fact that batches correspond to sequencing technologies the models encountered during pre-training, rather than donor- or laboratory-specific effects.

Pre-trained scGPT embeddings showed lower batch effects than those of the model fine-tuned on the specific dataset. Fine-tuning without correction caused the model to internalize batch effects, since doing so improved its ability to minimize the training loss for gene expression prediction.

Finally, we compared the effects of deactivating different numbers of features using steering versus randomly selected features (Figure 4 and Appendix Figure 9). For the pancreas dataset, performance increased for up to 25 deactivated features and then plateaued. In contrast, for the lung and the immune datasets, performance increased for up to 30 and 60 deactivated features, respectively, but then decreased significantly beyond these thresholds. Furthermore, we evaluated different clamping values in Appendix Figure 11. The ablation demonstrates that preservation of biological signal decreases with lower clamping values, while batch correction and total score reach their highest values for -2.

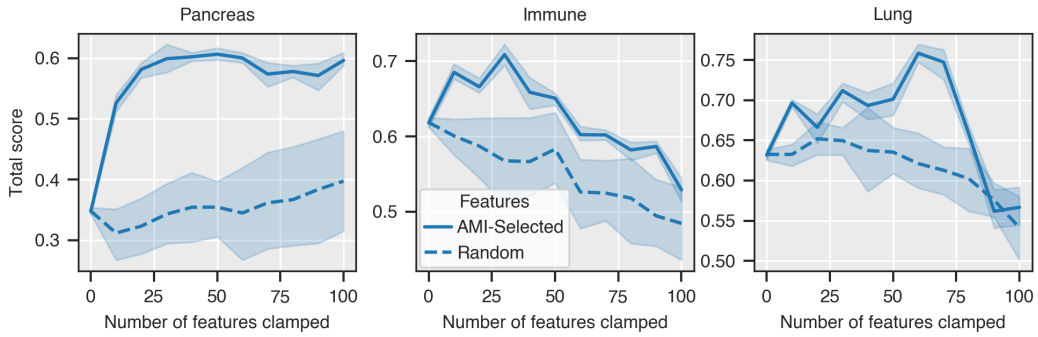

Figure 4: Total score (combining batch correction and biological conservation) as features are sequentially steered, selected randomly or by maximum AMI, across three datasets. Lines represent the mean across five seeds, and shaded regions indicate standard deviation.

5 Discussion & Conclusion

scFMs learn meaningful biological representations during pre-training. The features learned by the SAE are unsupervised, with the sole learning objective being token reconstruction. Some features will naturally capture frequently occurring genes or gene groups simply to improve reconstruction loss, rather than because the model assigns them functional relevance. However, we observe many features that are interpretable and biologically meaningful. Cell type concepts are varied and often achieve high AMI and F1 scores, even before downstream fine-tuning. This suggests that the models develop meaningful representations of cell biology during pre-training. While current studies report that scFMs underperform in strict zero-shot settings, the presence of rich biological features suggests that continued model development may close this gap and lead to foundation models with stronger zero-shot performance. However, a substantial proportion of features remains difficult to characterize, as standard methods such as Gene Ontology analysis prove insufficient for interpretation. Instead, these features often reflect heuristic gene co-expression patterns whose biological relevance is challenging to establish.

Table 2: Batch correction performance of single-cell foundation models on the pancreas dataset. Values show batch correction, biological conservation, and total scores, with higher scores indicating better performance. Means and standard deviations are computed across five runs with different random model initializations. Bold values highlight the best method within each category.

Method	Batch correction	Bio conservation	Total
Unintegrated	$0.13_{\pm 0.000}$	$0.67_{\pm 0.015}$	$0.45_{\pm 0.009}$
scVI	$0.86_{\pm 0.027}$	$0.98_{\pm 0.015}$	$\mathbf{0.93}_{\pm 0.016}$
scFoundation (zero-shot)	$0.34_{\pm 0.000}$	$0.43_{\pm 0.010}$	$0.39_{\pm 0.006}$
scFoundation (zero-shot, steered)	$0.49_{\pm 0.000}$	$0.36_{\pm 0.002}$	$\mathbf{0.41}_{\pm 0.001}$
scGPT (zero-shot)	$0.60_{\pm 0.017}$	$0.05_{\pm 0.011}$	$0.27_{\pm 0.011}$
scGPT (zero-shot, steered)	$0.67_{\pm 0.030}$	$0.09_{\pm 0.013}$	$\mathbf{0.32}_{\pm 0.015}$
scGPT (fine-tuned)	$0.43_{\pm 0.007}$	$0.54_{\pm 0.005}$	$0.50_{\pm 0.006}$
scGPT (fine-tuned, DAR)	$0.60_{\pm 0.034}$	$0.69_{\pm 0.005}$	$0.65_{\pm 0.012}$
scGPT (fine-tuned, steered)	$0.72_{\pm 0.038}$	$0.67_{\pm 0.005}$	$\mathbf{0.69}_{\pm 0.013}$

Training procedures affect information encoding. Different training protocols cause scFMs to learn differently, which affects the features detected by SAEs. Two key differences between scGPT and scFoundation features relate to gene expression and cell type diversity.

The lack of strong high expression features in scFoundation likely reflects how expression values are encoded. scGPT bins expression values, allowing the model to consistently map a bin to high expression. In contrast, scFoundation does not bin values and applies Bayesian downsampling during training. This simulates variance in sequencing read depth from different techniques and laboratories. As a result, high expression values do not always indicate that a gene is highly expressed relative to co-expressed genes. scGPT also produces more diverse cell type features, such as focusing on different gene subsets and expression patterns. In contrast, scFoundation typically focuses on 1-2 more simple features per cell type. This difference may result from training decisions such as higher masking percentages and fewer genes per sequence during pre-training, which may encourage scGPT to infer cell context from a wider variety of genes.

Cell type representations may lack cross-study integration The limited generalization of cell type features across all studies may partly reflect SAE limitations and how learned representations are decomposed, but could also indicate that models struggle to link cell types across different studies. This pattern could suggest that the underlying model representations do not fully consolidate cell types across experimental contexts.

Because pre-training is fully label-free, strong batch effects can cause the model to treat the same cell type in different studies as distinct types, preventing consolidation into unified concepts. The model learns to distinguish cells based on the patterns it observes in gene expression, but these patterns can be heavily influenced by technical factors such as sequencing protocols, laboratory conditions, or sample preparation methods. When these technical effects are strong enough, they may overshadow the shared biological signals that define a cell type, leading the model to learn separate representations for what should be the same biological concept. Fine-tuning scFMs may currently be essential not because it adds new information, but because it helps recontextualize existing information and bridge the gaps between study-specific representations. This recontextualization process may allow the model to recognize that cell types it learned separately are actually instances of the same biological concept.

SAE features are causally related to model behavior Steering provides causal evidence that features with high AMI scores relative to labels actually encode that information within the model. By selectively removing features and observing changes in model behavior, we demonstrate that these features are not merely correlated with biological concepts but actively contribute to how the model processes and represents cellular information. The fact that removing batch-related features improves batch integration metrics confirms that SAE features capture functionally relevant aspects of the model's internal representations.

The steering results also demonstrate that model representations are decomposable, meaning removing specific features can selectively alter model behavior without completely disrupting performance on other tasks. This modularity could be valuable for understanding and controlling model behavior in single-cell applications.

While steering is probably not useful as a batch correction method in its current form, the approach opens possibilities for more sophisticated interventions. There may be ways to explicitly encode batch information into models during training to enable its targeted removal. The ability to identify and manipulate specific features also suggests potential applications in selectively removing unwanted biases or technical artifacts from pre-trained models, similar to concept editing approaches in large language models where specific learned concepts can be modified without retraining the entire model.

Conclusion This work explored the use of sparse autoencoders to interpret the inner workings of single-cell foundation models. While the method shows potential, applying SAEs in the scRNA-seq context proves substantially more challenging than in language models, with feature interpretation requiring extensive manual effort and automated approaches showing limited success. As single-cell foundation models remain in early development without consolidated best practices, further interpretability studies examining what information these models encode, how informative their representations are, and the effects of different training protocols and architectural choices will be essential for advancing the field toward more reliable and controllable models.

REFERENCES

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

Abhinav Adduri, Dhruv Gautam, Beatrice Bevilacqua, Alishba Imran, Rohan Shah, Mohsen Naghipourfar, Noam Teyssier, Rajesh Ilango, Sanjay Nagaraj, Chiara Ricci-Tam, et al. Predicting cellular responses to perturbation across diverse contexts with state. *bioRxiv*, pp. 2025–06, 2025.

Constantin Ahlmann-Eltze, Wolfgang Huber, and Simon Anders. Deep learning-based predictions of gene perturbation effects do not yet outperform simple linear baselines. *BioRxiv*, pp. 2024–09, 2024.

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

Charlotte Bunne, Yusuf Roohani, Yanay Rosen, Ankit Gupta, Xikun Zhang, Marcel Roed, Theo Alexandrov, Mohammed AlQuraishi, Patricia Brennan, Daniel B Burkhardt, et al. How to build the virtual cell with artificial intelligence: Priorities and opportunities. *Cell*, 187(25):7045–7063, 2024.

Bart Bussmann, Patrick Leask, and Neel Nanda. Batchtopk sparse autoencoders. *arXiv preprint arXiv:2412.06410*, 2024.

Bart Bussmann, Noa Nabeshima, Adam Karvonen, and Neel Nanda. Learning multi-level features with matryoshka sparse autoencoders, 2025. URL https://arxiv.org/abs/2503.17547.

David Chanin, James Wilken-Smith, Tomáš Dulka, Hardik Bhatnagar, and Joseph Bloom. A is for absorption: Studying feature splitting and absorption in sparse autoencoders, 2024. URL https://arxiv.org/abs/2409.14507.

Charlotte Claye, Pierre Marschall, Wassila Ouerdane, CELINE HUDELOT, and Julien Duquesne. A framework to extract and interpret biological concepts from scRNAseq generative foundation

models. In *ICML 2025 Generative AI and Biology (GenBio) Workshop*, 2025. URL https://openreview.net/forum?id=wZpxbCwEe4.

Gerold Csendes, Kristóf Z Szalay, and Bence Szalai. Benchmarking a foundational cell model for post-perturbation rnaseq prediction. *bioRxiv*, pp. 2024–09, 2024. doi: 10.1101/2024.09.30.615843.

Haotian Cui, Chloe Wang, Hassaan Maan, Kuan Pang, Fengning Luo, Nan Duan, and Bo Wang. scgpt: toward building a foundation model for single-cell multi-omics using generative ai. *Nature methods*, 21(8):1470–1480, 2024.

Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. *arXiv preprint arXiv:2309.08600*, 2023.

Oscar Franzén, Li-Ming Gan, and Johan L M Björkegren. Panglaodb: a web server for exploration of mouse and human single-cell rna sequencing data, 2019.

Shi Fu, Yuzhu Chen, Yingjie Wang, and Dacheng Tao. A theoretical survey on foundation models, 2024. URL https://arxiv.org/abs/2410.11444.

Haoxiang Guan, Jiyan He, and Jie Zhang. Sparse autoencoders reveal interpretable structure in small gene language models. *arXiv preprint arXiv:2507.07486*, 2025.

Minsheng Hao, Jing Gong, Xin Zeng, Chiming Liu, Yucheng Guo, Xingyi Cheng, Taifeng Wang, Jianzhu Ma, Xuegong Zhang, and Le Song. Large-scale foundation model on single-cell transcriptomics. *Nature methods*, 21(8):1481–1491, 2024.

Fei He, Ruixin Fei, Jordan E Krull, Xinyu Zhang, Mingyue Gao, Li Su, Yibo Chen, Yang Yu, Jinpu Li, Baichuan Jin, et al. Harnessing the power of single-cell large language models with parameter efficient fine-tuning using scpeft. *bioRxiv*, pp. 2025–04, 2025.

John Healy and Leland McInnes. Uniform manifold approximation and projection. *Nature Reviews Methods Primers*, 4(1):82, 2024.

Graham Heimberg, Tony Kuo, Daryle J DePianto, Omar Salem, Tobias Heigl, Nathaniel Diamant, Gabriele Scalia, Tommaso Biancalani, Shannon J Turley, Jason R Rock, et al. A cell atlas foundation model for scalable search of similar human cells. *Nature*, 638(8052):1085–1094, 2025.

Kasia Z Kedzierska, Lorin Crawford, Ava P Amini, and Alex X Lu. Assessing the limits of zero-shot foundation models in single-cell biology. *BioRxiv*, pp. 2023–10, 2023.

Eric Kernfeld, Yunxiao Yang, Joshua S. Weinstock, Alexis Battle, and Patrick Cahan. A systematic comparison of computational methods for expression forecasting. *bioRxiv*, 2024. doi: 10.1101/2023.07.28.551039.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Arthur Liberzon, Chet Birger, Helga Thorvaldsdóttir, Mahmoud Ghandi, Jill P Mesirov, and Pablo Tamayo. The molecular signatures database (MSigDB) hallmark gene set collection. *Cell Syst*, 1 (6):417–425, December 2015.

Tianyu Liu, Kexing Li, Yuge Wang, Hongyu Li, and Hongyu Zhao. Evaluating the utilities of foundation models in single-cell data analysis. *bioRxiv*, 2024. doi: 10.1101/2023.09.08.555192. URL https://www.biorxiv.org/content/early/2024/12/10/2023.09.08.555192.

Romain Lopez, Jeffrey Regier, Michael B Cole, Michael I Jordan, and Nir Yosef. Deep generative modeling for single-cell transcriptomics. *Nature methods*, 15(12):1053–1058, 2018.

Malte D Luecken, Maren Büttner, Kridsadakorn Chaichoompu, Anna Danese, Marta Interlandi, Michaela F Müller, Daniel C Strobl, Luke Zappia, Martin Dugas, Maria Colomé-Tatché, et al. Benchmarking atlas-level data integration in single-cell genomics. *Nature Methods*, 19(1):41–50, 2022.

Olga Ovcharenko, Florian Barkmann, Philip Toma, Imant Daunhawer, Julia E Vogt, Sebastian Schelter, and Valentina Boeva. scssl-bench: Benchmarking self-supervised learning for single-cell data. *Forty-second International Conference on Machine Learning*, 2025.

CZI Cell Science Program, Shibla Abdulla, Brian Aevermann, Pedro Assis, Seve Badajoz, Sidney M Bell, Emanuele Bezzi, Batuhan Cakir, Jim Chaffer, Signe Chambers, et al. Cz cellxgene discover: a single-cell data platform for scalable exploration, analysis and modeling of aggregated data. *Nucleic acids research*, 53(D1):D886–D900, 2025.

Senthooran Rajamanoharan, Arthur Conmy, Lewis Smith, Tom Lieberum, Vikrant Varma, János Kramár, Rohin Shah, and Neel Nanda. Improving dictionary learning with gated sparse autoencoders, 2024. URL https://arxiv.org/abs/2404.16014.

Yanay Rosen, Yusuf Roohani, Ayush Agarwal, Leon Samotorčan, Tabula Sapiens Consortium, Stephen R Quake, and Jure Leskovec. Universal cell embeddings: A foundation model for cell biology. *bioRxiv*, pp. 2023–11, 2023.

Viktoria Schuster. Can sparse autoencoders make sense of gene expression latent variable models?, 2025. URL https://arxiv.org/abs/2410.11468.

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

Nicolas Steiner, Ziteng Li, Omid Vosoughi, Johanna Schrader, Soumyadeep Roy, Wolfgang Nejdl, and Ming Tang. A systematic evaluation of single-cell foundation models on cell-type classification task. WSDM '25, pp. 1112–1113, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 9798400713293. doi: 10.1145/3701551.3708811. URL https://doi.org/10.1145/3701551.3708811.

Aravind Subramanian, Pablo Tamayo, Vamsi K. Mootha, Sayan Mukherjee, Benjamin L. Ebert, Michael A. Gillette, Amanda Paulovich, Scott L. Pomeroy, Todd R. Golub, Eric S. Lander, and Jill P. Mesirov. Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles. *Proceedings of the National Academy of Sciences*, 102(43): 15545–15550, 2005. doi: 10.1073/pnas.0506580102.

Artur Szałata, Karin Hrovatin, Sören Becker, Alejandro Tejada-Lapuerta, Haotian Cui, Bo Wang, and Fabian J Theis. Transformers in single-cell omics: a review and new perspectives. *Nature methods*, 21(8):1430–1443, 2024.

Adly Templeton, Tom Conerly, Jonathan Marcus, Jack Lindsey, Trenton Bricken, Brian Chen, Adam Pearce, Craig Citro, Emmanuel Ameisen, Andy Jones, Hoagy Cunningham, Nicholas L Turner, Callum McDougall, Monte MacDiarmid, C. Daniel Freeman, Theodore R. Sumers, Edward Rees, Joshua Batson, Adam Jermyn, Shan Carter, Chris Olah, and Tom Henighan. Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet. *Transformer Circuits Thread*, 2024. URL https://transformer-circuits.pub/2024/scaling-monosemanticity/index.html.

Christina V Theodoris, Ling Xiao, Anant Chopra, Mark D Chaffin, Zeina R Al Sayed, Matthew C Hill, Helene Mantineo, Elizabeth M Brydon, Zexian Zeng, X Shirley Liu, et al. Transfer learning enables predictions in network biology. *Nature*, 618(7965):616–624, 2023.

Alexander Theus, Florian Barkmann, David Wissel, and Valentina Boeva. Cancerfoundation: A single-cell rna sequencing foundation model to decipher drug resistance in cancer. *bioRxiv*, pp. 2024–11, 2024.

Fan Yang, Wenchuan Wang, Fang Wang, Yuan Fang, Duyu Tang, Junzhou Huang, Hui Lu, and Jianhua Yao. scbert as a large-scale pretrained deep language model for cell type annotation of single-cell rna-seq data. *Nature Machine Intelligence*, 4(10):852–866, 10 2022. ISSN 2522-5839. doi: 10.1038/s42256-022-00534-z. URL https://doi.org/10.1038/s42256-022-00534-z.

Masahiro Yoshida, Kaylee B Worlock, Ni Huang, Rik GH Lindeboom, Colin R Butler, Natsuhiko Kumasaka, Cecilia Dominguez Conde, Lira Mamanova, Liam Bolt, Laura Richardson, et al. Local and systemic responses to sars-cov-2 infection in children and adults. *Nature*, 602(7896): 321–327, 2022.

A APPENDIX

A.1 DATASETS

All benchmark datasets utilized in our study are openly accessible to the public.

CellXGene Census: The Census provides an efficient tool to access and query all single-cell RNA data from CZ CELLxGENE Discover (Program et al., 2025). By querying for healthy cells across a range of tissues, we obtained a dataset of over 37 million cells. This dataset was reportedly used by the authors of scGPT for model training.

COVID-19: The COVID-19 dataset (Yoshida et al., 2022) includes 33,105 genes measured in 422,220 peripheral blood mononuclear cells, covering 16 annotated cell types from both healthy individuals and COVID-19 patients. Availability: `https://datasets.cellxgene.cziscience.com/ae49598b-646d-4325-b3e7-b164ac49d506.h5ad`

Immune: This collection consists of 33,506 cells containing 12,303 genes sourced from ten distinct donors, compiled by Luecken et al.(2022) across five research studies. While one investigation obtained cells from human bone marrow, the remaining four studies extracted cells from human peripheral blood. The dataset contains annotations for 16 distinct cell types. Availability: `https://doi.org/10.6084/m9.figshare.12420968.v8`

Pancreas: This collection was reprocessed by Luecken et al.(2022) through the integration of five human pancreas studies. It encompasses 16,382 cells, featuring 19,093 genes, sequenced using four scRNA-seq platforms (inDrop, CEL-Seq, Smart-Seq2, SMARTer). The integrated dataset incorporates 14 cell types (alpha, beta, gamma, ductal, acinar, delta, pancreatic stellate, pancreatic polypeptide, endothelial, macrophage, mast, epsilon, Schwann and T cell). Availability: `https://figshare.com/ndownloader/files/24539828`

Lung: This collection encompasses 32,426 cells spanning 16 batches and two sequencing platforms (Drop-seq and 10x Chromium), compiled by Luecken et al.(2022) from three research laboratories. The integrated dataset incorporates 15,148 genes. The cells originate from transplant patients and lung tissue samples and are classified into 17 cell types. Availability: `https://figshare.com/ndownloader/files/24539942`

A.2 BATCH CORRECTION METRICS

To evaluate how well different methods integrate cells from different experimental batches, we follow Luecken et al. (2022). The evaluation metrics are organized into two distinct groups: one set focuses on assessing how well the biological variability is preserved, while the other assesses the effectiveness of aligning cells from different batches.

For assessing the preservation of biological variation, we employ several metrics, including the isolated labels score, normalized mutual information (NMI) and adjusted rand index (ARI), silhouette label score, and the cLISI metric. For measuring batch correction performance, we utilize graph connectivity analysis, kBET calculations per label, individual cell iLISI values, PCR comparison scores, and batch-specific silhouette coefficients.

The bio conservation and batch correction scores are computed by first min-max normalizing each individual metric and then taking the mean across all bio conservation or batch correction metrics, respectively. The total score is calculated as $0.6\times$ bio conservation $+ 0.4\times$ batch correction. Comprehensive descriptions of these metrics can be found in Luecken et al. (2022).

A.3 SPARSE AUTOENCODERS

A.3.1 ARCHITECTURES

Sparse autoencoders are a relatively new approach to discovering interpretable features in foundation models. Several methods exist for applying a sparsity penalty to the feature space. The early version by Bricken et al. (2023) simply applies an L1 regularization penalty to the feature activations. Gated sparse autoencoders (Rajamanoharan et al., 2024) decouple the detection of which features are active from the estimation of their magnitudes. BatchTopK sparse autoencoders (Bussmann et al.,

2024) use an activation function that retains only the k largest latents per batch, discarding the L1 regularization entirely. Matryoshka sparse autoencoders (Bussmann et al., 2025) do not address the sparsity penalty, but instead introduce a hierarchical feature structure by training multiple nested dictionaries of increasing size. Smaller dictionaries are forced to independently reconstruct the inputs without relying on the larger ones. The aim is to reduce the incentive created by the sparsity penalty for more specific concepts to absorb high-level features (Chanin et al., 2024).

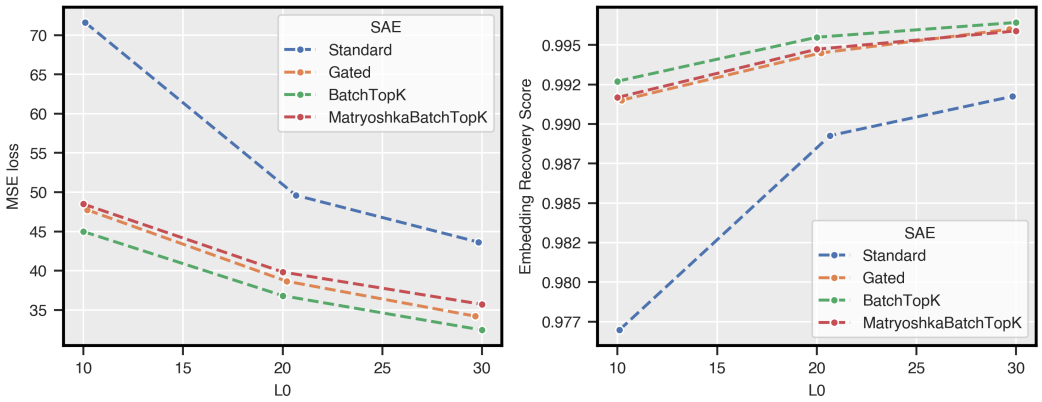

Figure 5: Performance of the four different SAE architectures at different sparsity levels.

A.3.2 Embedding Recovery Score

The Embedding Recovery Score provides an estimate on the impact of token-level reconstruction downstream, by comparing the cell embeddings produced by the model in different scenarios.

- X_{original} is the embedding from the original input,
- $X_{\text{reconstructed}}$ is the embedding after reconstructing the tokens,
- X_0 is the embedding when all tokens are zeroed out.

The Embedding Recovery Score is then defined as:

$$\text{Embedding Recovery Score} = \frac{\text{MSE}(X_0, X_{\text{original}})}{\text{MSE}(X_{\text{original}}, X_{\text{reconstructed}})}$$

A.3.3 Training Hyperparameters

All sparse autoencoders were trained by sampling residual stream activations without replacement from the scFM with a batch size of 8,192. The SAE architecture used for all experiments was a one hidden layer MLP with BatchTopK sparsity restriction.

Hyperparameters were adjusted according to dataset size and complexity as seen in Table 3. Smaller datasets (Pancreas, Lung, Immune) used a higher learning rate of 1e-3, while larger datasets (COVID-19, CellxGene) used a lower learning rate of 1e-4 for more stable training. The CellxGene dataset used a larger latent dimension (1024 vs 512) and higher sparsity value (20 vs 10) to accommodate its increased variability and provide better expressiveness. The sparsity term refers to the top k active neurons per batch as described by Bussmann et al. (2024).

Table 3: Sparse autoencoder hyperparameters for each dataset.

Dataset	Learning Rate	Latent Dimension	Sparsity
Pancreas	0.001	512	10
Lung	0.001	512	10
Immune	0.001	512	10
COVID-19	0.0001	512	10
CellxGene	0.0001	1024	20

A.4 FEATURE CHARACTERIZATION

This appendix provides detailed examples of interpretable features discovered by sparse autoencoders across different models and datasets. Features are organized into two main categories as described in Section **??**: gene-specific features that reflect properties of individual genes, and cell-specific features that capture properties of entire cells through contextual information learned by the transformer models.

A.4.1 GENE-SPECIFIC FEATURES

Expression Level
These features activate within specific ranges of gene expression values. To quantify this relationship, we computed the Spearman rank correlation between feature activations and input gene expression vectors, after centering by the mean expression across all positions where the feature is active. Note that scGPT uses binned expression values (1-51), while scFoundation uses continuous values (0-8.77 for the COVID-19 dataset).

Table 4: Expression level features in pre-trained scGPT (COVID-19 dataset)

Feature	Expression (mean $\pm$ SD)	Spearman correlation	Density
403	41.38 ± 4.76	0.58	0.18
227	32.16 ± 6.37	0.41	0.15
398	14.23 ± 6.55	0.44	0.16
92	5.48 ± 4.38	0.65	0.18

Table 5: Expression level features in pre-trained scFoundation (COVID-19 dataset)

Feature	Expression (mean $\pm$ SD)	Spearman correlation	Density
482	3.64 ± 0.82	0.38	0.1
451	1.26 ± 0.41	0.74	0.39

Gene family
These are features that activate specifically for genes belonging to particular functional families. Feature activations were thresholded at 0.3 to identify the most specific patterns.

Table 6: Gene family features in pre-trained scGPT (COVID-19 dataset).

Feature	Gene family	Percentage (%)	Unique Genes
287	Immunoglobulins	92	57
334	Metallothioneins	100	4
276	Histones	95	26
302	T cell receptors	40	41
262	Human leukocyte antigens	99	13
172	Mitochondrially encoded protein genes	100	25
181	Ribosomal protein genes	95	79

Table 7: Gene family features in pre-trained scFoundation (COVID-19 dataset).

Feature	Gene family	Percentage (%)	Unique Genes
260	Human leukocyte antigens	100	5
333	Mitochondrially encoded protein genes	100	5
423	Ribosomal protein genes	76	160

Biological process
These features capture genes involved in specific biological processes through functional gene modules. Feature activations were thresholded at above 0.5 to identify the strongest patterns, and enrichment is reported as the adjusted p-value.

Table 8: Biological process features in pre-trained scGPT (COVID-19 dataset).

Feature	Biological process	Adj. p-value
50	Nuclear Outer Membrane–ER Network	9.00e-48
173	Cell Cycle	1.70e-92
213	Phagocytosis	8.43e-18
233	Hemostasis	1.52e-21
330	Adaptive Immune Response	5.56e-60

Table 9: Biological process features in pre-trained scGPT. Here the SAE was trained on the Cellx-Gene Census and evaluated across CellxGene and COVID-19 datasets.

Feature	Biological process	Adj. p-value for COVID-19	Adj. p-value for CellxGene
114	Cell Cycle	2.58e-88	1.86e-210
404	ER to Golgi Transport	2.101e-34	1.12e-73
765	Programmed Cell Death	3.27e-12	3.73e-79
816	Response to Virus	1.88e-27	1.75e-96

A.4.2 CELL SPECIFIC

Cell type
These features correspond to specific cell types, evaluated using Adjusted Mutual Information (AMI) and F1 scores with optimized activation thresholds per cell type.

Table 10: Cell type features in pre-trained scGPT (COVID-19 dataset)

Feature	Cell type	AMI	F1
201	T cells	0.47	0.86
119	T naive cells	0.60	0.81
151	B cells	0.99	1.00
492	B memory cells	0.54	0.77
420	Monocytes	0.64	0.83
57	Monocytes or dendrtitic cells	0.67	0.88
59	NK cells	0.44	0.69
145	Hematopoietic progenitor cells	1.00	1.00
223	Platelets	0.97	0.99
462	Plasma cells	1.00	1.00

Table 11: Cell type features in pre-trained scFoundation (COVID-19 dataset)

Feature	Cell type	AMI	F1
266	T cells	0.32	0.79
194	T naive cells	0.56	0.79
190	B cells	0.93	0.98
445	B memory cells	0.50	0.75
277	Monocytes	0.58	0.80
230	Monocytes or dendritic cells	0.61	0.83
380	NK cells	0.34	0.62
508	Hematopoietic progenitor cells	0.30	0.50
116	Platelets	0.88	0.95
337	Plasma cells	0.51	0.69

Disease

Features that preferentially activate in cells from patients with specific disease conditions, evaluated using Adjusted Mutual Information (AMI) and F1 scores with optimized activation thresholds.

Table 12: COVID-19 related features in pre-trained scGPT (COVID-19 dataset)

Feature	Disease	AMI	F1
127	COVID-19	0.11	0.59
89	post-COVID-19 disorder	0.20	0.55

Table 13: COVID-19 related features in pre-trained scFoundation (COVID-19 dataset)

Feature	Disease	AMI	F1
186	COVID-19	0.21	0.63
250	post-COVID-19 disorder	0.18	0.50

Sequencing Technology

Features that activate based on technical aspects of the data, including sequencing technologies and experimental protocols, evaluated using Adjusted Mutual Information (AMI) and F1 scores with optimized activation thresholds.

Table 14: Sequencing technology features in pre-trained scGPT (Pancreas dataset)

Feature	Technology	AMI	F1
327	smartseq2	0.68	0.86
366	celseq2	0.57	0.80
307	smarter	0.86	0.95
366	celseq	0.38	0.74
429	fluidigmc1	0.87	0.94

Table 15: Sequencing technology features in pre-trained scFoundation (Pancreas dataset)

Feature	Technology	AMI	F1
69	smartseq2	0.74	0.92
63	celseq2	0.70	0.88
200	smarter	0.56	0.76
261	fluidigmc1	0.57	0.77

A.4.3 SPECIFIC EXAMPLES

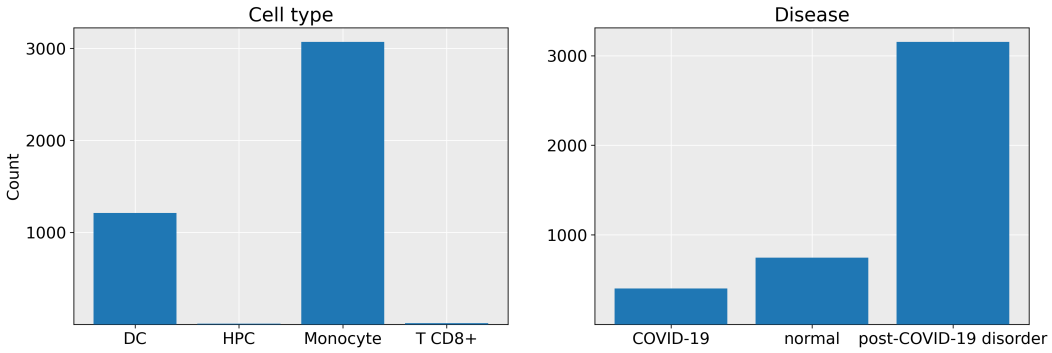

Figure 6: Distribution of token activations of featue 224 across cell types and COVID-19 status. Feature activations were thresholded at above 0.5

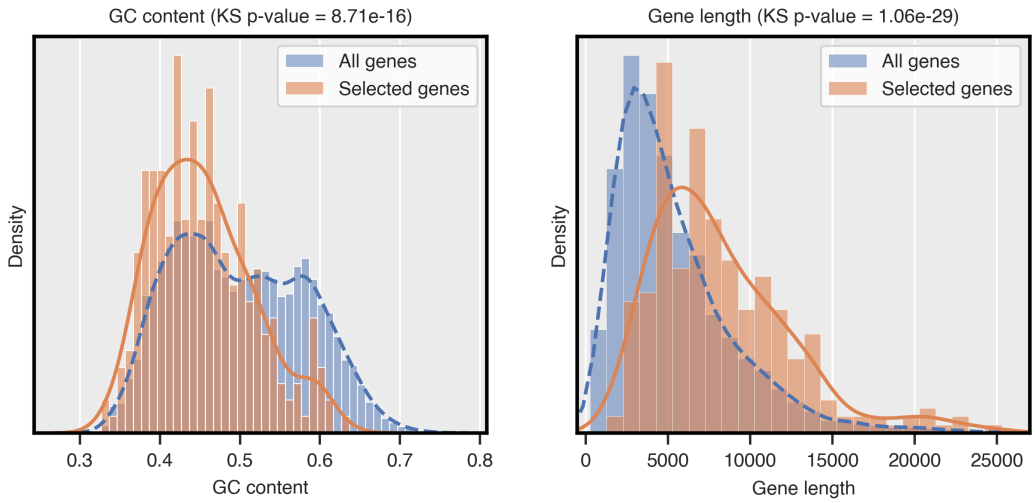

Figure 7: Gene GC content and length distribution for all genes in the dataset (blue) and those with highest activation in feature 236 (orange) for the pre-trained scGPT model on the pancreas dataset.

A.5 FEATURE STEERING

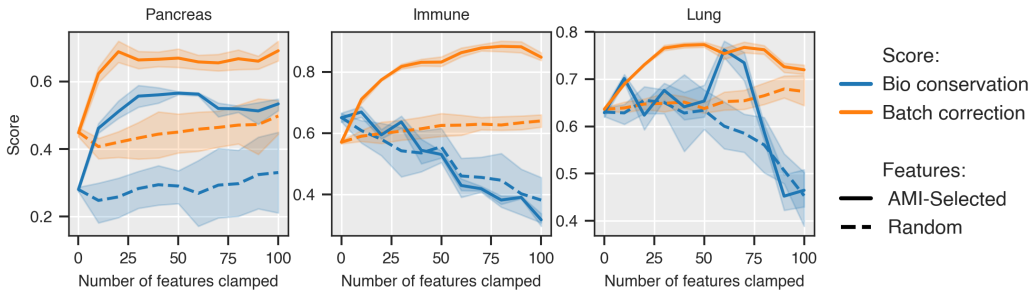

Figure 9: Biological conservation and batch correction scores as features are sequentially steered, selected randomly or by maximum AMI, across three datasets. Lines show the mean over five seeds, and shaded regions indicate standard deviation.

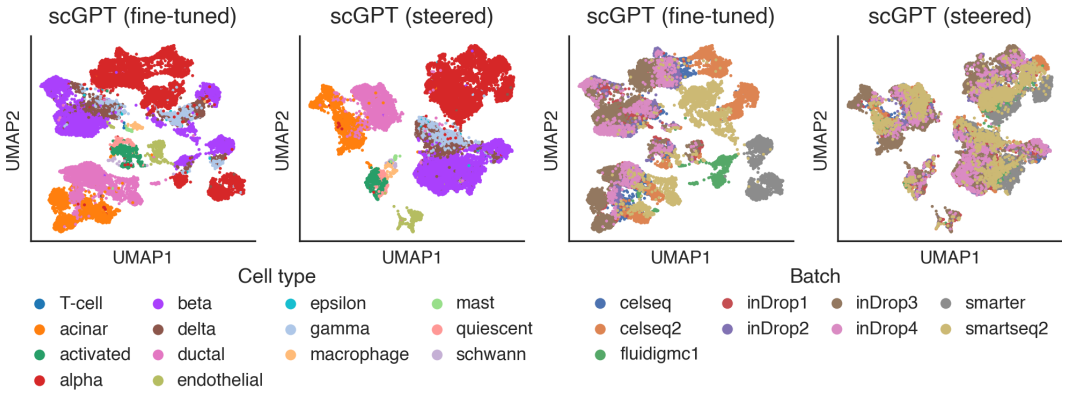

Figure 8: UMAP of the standard and steered cell embeddings of fine-tuned scGPT, colored by cell type (left) and sequencing protocol (right).

Table 16: Batch correction performance of single-cell foundation models on the lung dataset. Values show batch correction, biological conservation, and total scores, with higher scores indicating better performance. Means and standard deviations are computed across five runs with different random model initializations. Bold values highlight the best method within each category.

Method	Batch correction	Bio conservation	Total
Unintegrated	0.12±0.000	0.92±0.017	0.60±0.010
scVI	0.88±0.013	0.72±0.008	**0.79**±0.005
scFoundation (zero-shot)	0.50±0.000	0.80±0.003	0.68±0.002
scFoundation (zero-shot, steered)	0.48±0.000	0.84±0.009	**0.69**±0.005
scGPT (zero-shot)	0.66±0.009	0.10±0.003	**0.32**±0.004
scGPT (zero-shot, steered)	0.56±0.029	0.01±0.003	0.23±0.012
scGPT (fine-tuned)	0.55±0.005	0.85±0.019	0.73±0.013
scGPT (fine-tuned, DAR)	0.66±0.004	0.84±0.006	0.77±0.004
scGPT (fine-tuned, steered)	0.72±0.005	0.88±0.011	**0.82**±0.007

Table 17: Batch correction performance of single-cell foundation models on the immune dataset. Values show batch correction, biological conservation, and total scores, with higher scores indicating better performance. Means and standard deviations are computed across five runs with different random model initializations. Bold values highlight the best method within each category.

Method	Batch correction	Bio conservation	Total
Unintegrated	0.09±0.000	0.67±0.004	0.44±0.002
scVI	0.73±0.009	0.88±0.015	**0.82**±0.010
scFoundation (zero-shot)	0.46±0.010	0.70±0.051	**0.60**±0.035
scFoundation (zero-shot, steered)	0.43±0.024	0.50±0.116	0.47±0.060
scGPT (zero-shot)	0.66±0.011	0.14±0.005	**0.34**±0.005
scGPT (zero-shot, steered)	0.78±0.018	0.03±0.016	0.33±0.012
scGPT (fine-tuned)	0.51±0.005	0.77±0.010	0.67±0.005
scGPT (fine-tuned, DAR)	0.75±0.005	0.73±0.017	0.74±0.011
scGPT (fine-tuned, steered)	0.72±0.010	0.76±0.012	**0.75**±0.005

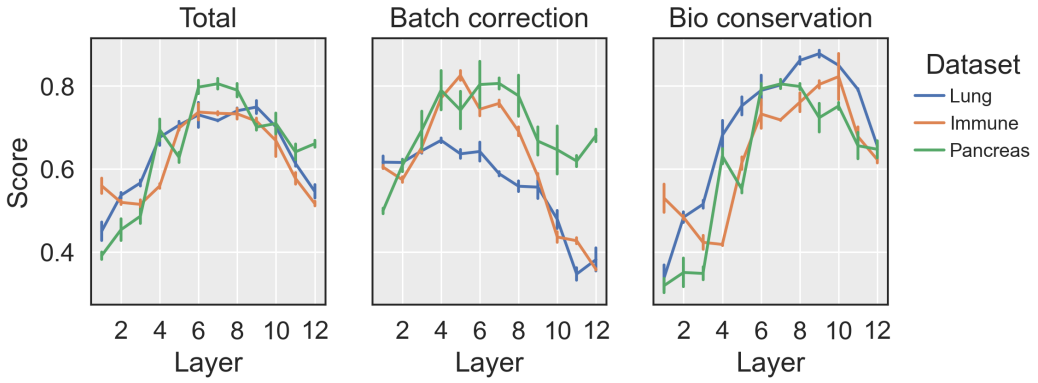

Figure 10: Layer-wise performance of SAE features for batch correction. Performance metrics (total score, batch correction, and biological conservation) are shown for features extracted from different layers (1-12) of scGPT across three datasets: Lung (blue), Immune (orange), and Pancreas (green). Error bars represent standard deviation across five random seeds.

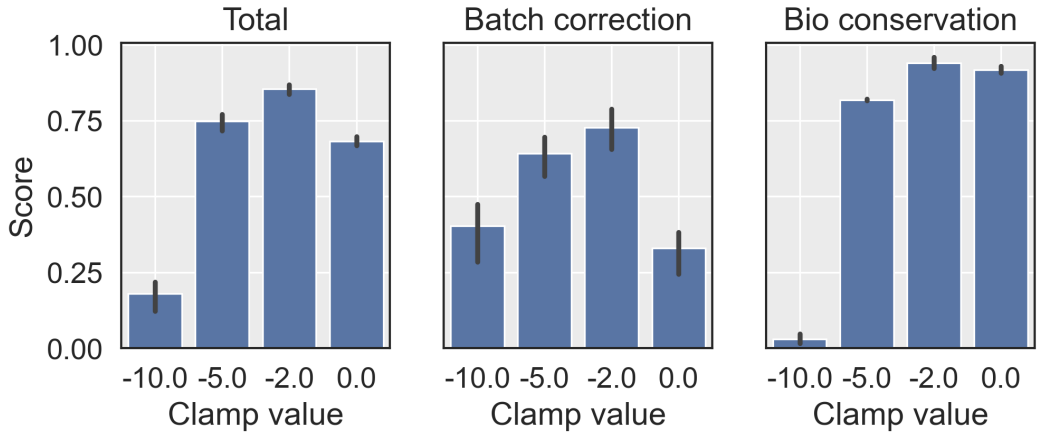

Figure 11: Effect of different clamping values on batch correction performance. Comparison of different clamping values (-10.0, -5.0, -2.0, 0.0) on three evaluation metrics: total score (left), batch correction score (middle), and biological conservation score (right). Bar heights represent mean scores across datasets, with error bars indicating standard deviation.

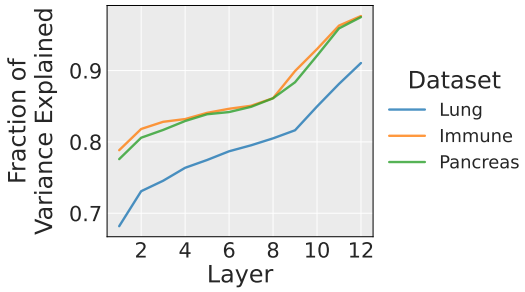

Figure 12: The fraction of variance explained for SAEs trained on different layers of scGPT across three datasets: Lung (blue), Immune (orange), and Pancreas (green).

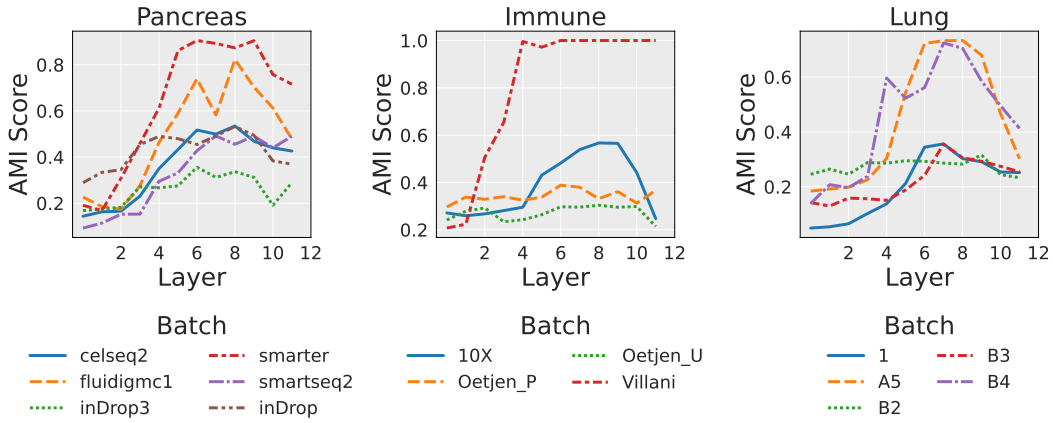

Figure 13: Layer-wise analysis of batch feature emergence in scGPT. For each layer, we extracted the top three features most associated with each batch effect and computed their mean AMI score. Each line represents a different batch variable of the corresponding dataset.

A.6 Disclosure of LLM Editing Tools

This paper was edited for clarity using large language models. All scientific content is the authors' own.