# SPARSE AUTOENCODERS REVEAL INTERPRETABLE FEATURES IN SINGLE-CELL FOUNDATION MODELS

**Flavia Pedrocchi**[*], **Florian Barkmann**[*], **Amir Joudaki**[†] **& Valentina Boeva**[†]
Department of Computer Science
ETH Zurich
{ajoudaki,vboeva}@ethz.ch

## ABSTRACT

Single-cell foundation models (scFMs) hold promise for applications in cell type annotation, data integration, and prediction of the effects of cell perturbations, but their internal mechanisms remain poorly understood. We investigate the structure of these models by training sparse autoencoders (SAEs) on the hidden representations of three widely used scFMs: scGPT, scFoundation, and Geneformer. 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. Finally, we demonstrate that SAE-derived features are functionally 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 powerful tools in computational biology for analyzing 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 (Ahlmann-Eltze et al., 2026) and applied to downstream tasks including cell type annotation (Rosen et al., 2023; Heimberg et al., 2025), batch integration (He et al., 2025; Cui et al., 2024), and perturbation prediction such as drug response (Hao et al., 2024; Theus et al., 2024) and gene perturbation response (Adduri et al., 2025). Their promise lies in capturing complex, non-linear relationships in high-dimensional transcriptomics data and generalizing across 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 function as black boxes (Fu et al., 2024), making it unclear how predictions are made. This lack of transparency is problematic given their architectures are largely inherited from NLP models with minimal biological adaptation (Theodoris et al., 2023; Yang et al., 2022). Recent benchmarking reveals mixed performance, with linear models sometimes matching or outperforming scFMs in cell type classification (Kedzierska et al., 2023) and perturbation prediction (Ahlmann-Eltze et al., 2024; Csendes et al., 2024; Kernfeld et al., 2024), though results are task- and dataset-dependent. Additionally, scFMs often require fine-tuning for practical usability (Liu et al., 2024; Steiner et al., 2025; Ovcharenko et al., 2025). Understanding 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

[*]Co-first authors
[†]Equal last authors

language models (Guan et al., 2025), and protein language models (Simon & Zou, 2024; Adams et al., 2025; Gujral 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 can have a complex and meaningful understanding of cell biology. 2) We show how model architectures and training procedures can affect the encoding of information within the model. 3) We characterize how pre-trained scFMs represent technical variation alongside biological information, finding that cell type features often show study-specific patterns rather than consistent activation across all datasets. 4) We demonstrate that SAE-derived features are functionally related to model behavior and can be intervened upon to improve batch integration in scFMs. 5) We release a codebase for training sparse autoencoders on single-cell foundation models that is easily extensible to new autoencoder architectures and foundation models.[1]

## 2 Background

### 2.1 Sparse autoencoders

Sparse autoencoders (SAEs) are neural networks that learn interpretable representations by decomposing neural network activations into sparse, monosemantic features. These latent units respond to single, interpretable concepts rather than multiple unrelated patterns. Recent work has demonstrated their effectiveness for interpreting transformer models, from language models (Cunningham et al., 2023; Bricken et al., 2023; Templeton et al., 2024) to biological sequence models (Guan et al., 2025; Simon & Zou, 2024; Adams et al., 2025; Gujral et al., 2025).

### 2.2 Sparse autoencoders for single-cell foundation models

Recent work by Schuster (2025) and Claye et al. (2025) utilized Sparse Autoencoders (SAEs) to link biological concepts to the final cell embedding space. This study extends these efforts by applying SAEs to latent intermediate token representations during the model's forward pass, capturing richer biological information before it is compressed into cell-level summaries. By comparing three foundation models (scGPT, Geneformer, and scFoundation) across diverse datasets, we analyze how distinct training architectures influence learned representations. Furthermore, we categorize gene- and cell-specific features and introduce novel applications of SAE-based steering to address technical artifacts such as batch effects while preserving biological signals.

## 3 Methods

### 3.1 Sparse Autoencoder Training

We trained sparse autoencoders on intermediate token representations from three single-cell foundation models: scGPT (Cui et al., 2024), scFoundation (Hao et al., 2024), and Geneformer V2 (Theodoris et al., 2023). For each scFM, we performed inference on scRNA-seq datasets and extracted residual stream activations from different transformer layers (Figure 1).

**Model selection.** We analyzed pre-trained and fine-tuned variants where available. scFoundation lacked publicly available fine-tuning code, so we used only the pre-trained model. Pre-trained Geneformer produced poorly-defined features with high reconstruction loss (Figure 6), so we focused on fine-tuned Geneformer. We prioritize scGPT analysis due to its widespread adoption and fine-tuning ease.

**SAE architecture.** We used BatchTopK SAEs (Bussmann et al., 2024), which outperformed other SAE architectures in both language models and our preliminary experiments (Appendix A.4.1). We trained with Adam (Kingma & Ba, 2014), sampling 250 tokens per cell and constructing batches of 8,192 tokens drawn from varied cellular contexts following Bricken et al. (2023). We kept feature

[1] `https://github.com/BoevaLab/sparse-autoencoders-for-scfm`.

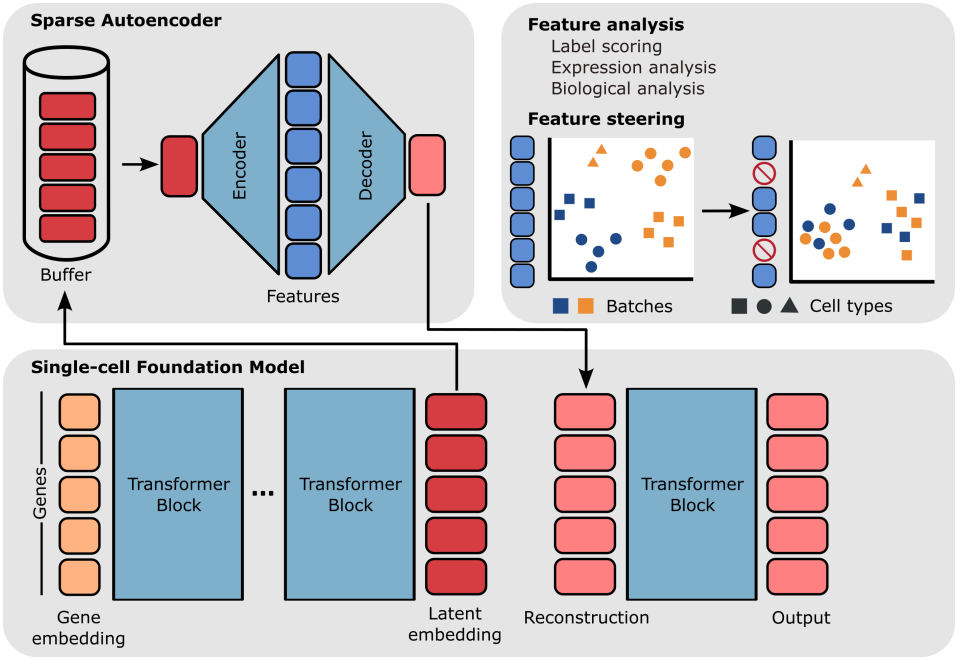


Figure 1: Framework used to train and evaluate sparse autoencoders from scFM latent gene embeddings.

spaces small, as larger dictionaries split concepts into overly fine-grained, uninterpretable features, especially on smaller datasets. Details on the SAE hyperparameters can be found in Appendix A.4.3.

**Layer selection.** We trained SAEs on all 12 transformer layers. Middle layers (5-8) achieved optimal steering performance and captured batch-related features most effectively (Figures 14, 16), while later layers (9-11) showed higher reconstruction quality and more structured biological representations (Figures 6, 11). We selected layer 6 for steering experiments and layer 10 for interpretability analysis, avoiding final layers which primarily encode prediction-specific features rather than generalizable representations.

**Evaluation.** The standard SAE metrics "loss recovered" (Bricken et al., 2023) failed due to consistently high scFM training loss. To address this limitation, we developed the Embedding Recovery Score, which measures how token-level reconstruction quality affects downstream cell embeddings (Appendix A.4.2).

### 3.2 Feature Analysis

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

**Cell-level associations:** SAEs produce continuous activation values for each feature at each gene position within a cell. We aggregated these gene-level activations to the cell level via max-pooling, yielding a single activation score per feature per cell. This aggregation allows us to ask: does this feature consistently activate for specific cell types, disease states, or technical batches?

To quantify these associations, we thresholded continuous cell-level scores into binary classifications (feature active/inactive) and computed alignment metrics, namely adjusted mutual information (AMI) and F1 scores, against ground-truth cell-level labels. We examined multiple threshold values since the interpretation of "activation" varies and often focused on strongly activated tokens to identify each feature's core concept. For example, a feature might activate weakly across many B cells but strongly in a specific B cell subtype, revealing finer-grained biological structure.

**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. We also used PanglaoDB marker sets (Franzén et al., 2019) to examine enrichment of cell type-specific expression markers.

Additionally, we computed Spearman's rank correlations to assess whether features consistently activated for tokens encoding high gene expression values, and quantified feature enrichment for specific gene families by calculating the fraction of each feature's activations attributed to genes within those families.

### 3.3 Feature Steering

We used steering 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. Specifically, we tested whether SAE features can be manipulated to reduce batch effects while preserving the biological signal.

**Procedure.** We identified batch-correlated features by computing adjusted mutual information (AMI) between feature activations and batch labels (Section 3.2). During inference, we passed data through the scFM to obtain token representations, projected these into feature space via the SAE encoder, then clamped identified "batch features" to -2 whenever they activated. We projected the modified activations back to token space via the SAE decoder and fed them to remaining scFM layers to generate new cell embeddings. We explored different clamp values (Figure 15) and -2 balanced batch correction against biological signal preservation.

**Evaluation.** We compared steered embeddings against: (1) original scFM embeddings, (2) PCA on raw gene counts (no batch removal), and (3) scVI (Lopez et al., 2018), a conditional VAE-based batch correction method. We assessed batch integration quality and biological conservation using metrics detailed in Appendix A.2. scVI serves as a reference point for batch effect magnitude rather than a competitive baseline.

### 3.4 Datasets

We trained SAEs on five datasets: CellXGene Census (37M cells) (CZI Cell Science Program et al., 2025), a COVID-19 cohort (Yoshida et al., 2022), and three tissue-specific datasets (immune, lung, pancreas) from Luecken et al. (2022). These three datasets were compiled for a batch integration benchmarking study and provide controlled scenarios with varying batch effects and cell type compositions that can be used for steering validation. COVID-19 and CellXGene Census enabled large-scale feature characterization. Details in Appendix A.1.

## 4 Results

### 4.1 Sparse Autoencoders find interpretable concepts

Sparse autoencoders trained on scGPT, scFoundation, and Geneformer revealed that scFMs organize information along two distinct axes: gene-specific features that encode properties of individual genes independent of cellular context, and cell-specific features that capture properties of entire cells, distributed across multiple gene tokens. This decomposition reveals how transformer architectures integrate local (gene) and global (cell) information during sequence processing.

**Gene-specific features encode expression levels, gene identity, and molecular function.** These features activate based on properties intrinsic to individual gene tokens. Across all three models, we identified features correlating with gene expression values, though the encoding strategy varied by architecture (Tables 5, 4, Figure 8). scFoundation features primarily captured low expression values, likely because this model applies Bayesian downsampling during training, making high expression values inconsistent indicators of relative abundance. scGPT showed stronger expression correlations across multiple expression levels due to its binning strategy, which maps expression

ranges to discrete bins and removes technology-based variability. Geneformer, which encodes expression through token position rather than token value, produced features that activated most often at specific sequence positions.

Beyond expression, gene-specific features captured gene identity and family membership. We found features selective for ribosomal, mitochondrial, HLA, and immunoglobulin gene families (Tables 6, 7, 8). A third category encoded biological processes such as cell cycle, defense response and apoptosis through activation on functionally related gene sets (Tables 9, 10, 11).

**Cell-specific features encode cell identity through distributed representations.** Unlike gene-specific features, these activate based on the broader cellular context in which a gene appears. They arise from contextual information that transformers propagate across the sequence during self-attention, often distributed across many tokens rather than concentrated in the most biologically relevant genes.

In the COVID-19 dataset, we identified features corresponding to major cell types across all models, with broader categories (e.g. monocytes, B cells, T cells) represented more consistently than fine-grained subtypes (Tables 12, 13, 14). These cell type features were often enriched for appropriate marker genes and biological processes. However, feature structure varied across models, with scGPT having substantially more numerous and diverse feature types per cell type than scFoundation or Geneformer (Figure 2). These cell type features could be categorized as marker gene selective, generalized, ribosomal, or low expression based on their activation patterns. scGPT produced 3-8 diverse features per cell type even before fine-tuning, while scFoundation most often generated 2-3 complementary features per cell type. Geneformer required fine-tuning to develop clear representations and showed predominantly generalized features.

Beyond cell types, we found features encoding disease status (activating preferentially in COVID-19 patients) and technical batch effects (activating for specific studies, donors, or sequencing technologies) (Tables 15, 16, 17, 18).

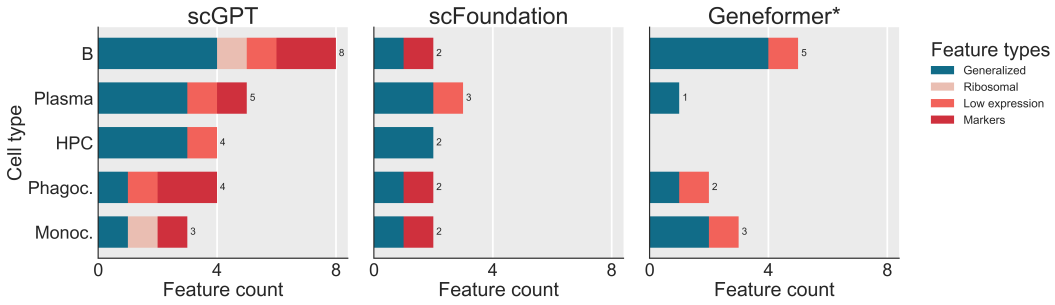


Figure 2: Distribution of feature types across the best represented cell types in the scFMs. Stacked bars show the number of features identified for each cell type, categorized by activation pattern: generalized features that activate broadly across genes and expression levels, marker features selective for canonical cell type markers, ribosomal features activating on ribosomal genes, and low expression features that preferentially activate on lowly expressed genes. Numbers indicate total features per cell type. *Geneformer needed to be fine-tuned before showing cell type-specific features while the other two models are pre-trained in this example.

### 4.2 Concepts from pre-trained scFM are varied and meaningful

The following findings demonstrate that pre-trained scFMs can develop rich internal representations even before task-specific fine-tuning. These representations are compositional, combining multiple distinct features to encode cell identity and transfer to contexts not present during training, such as disease states. We focus on scGPT, as it best exemplified these properties among the pre-trained models we analyzed.

**Features capture diverse aspects of cell identity.** For each cell type, we observed several distinct features with different characteristics, suggesting scFMs build cell representations from composi-

tional features rather than unified cell type detectors. Table 1 illustrates this using B-cell features from pre-trained scGPT.

Some features activated broadly across many genes and expression levels, representing general cell type membership (e.g., feature 478). Other features targeted specific marker genes: feature 363 activated strongly (activation $\geq 0.5$) on only 83 genes, 26 of these were known B-cell markers and another 38 immunoglobulin or MHC genes (both commonly enriched in B cells). Gene Ontology analysis confirmed enrichment for B-cell-mediated immunity (adjusted p-value = 2.1e-17). This feature thus captures a highly specific B-cell signature through canonical marker gene activity.

While these two features follow expected patterns by activating on canonical markers or broadly across B-cell contexts, we identified two additional unexpected encoding strategies. First, feature 151 implements *negative encoding*: it activates moderately on genes enriched for T-cell, monocyte, macrophage, megakaryocyte, and NK cell markers, but notably not B-cell markers. Rather than directly detecting B-cell signatures, this feature encodes "absence of non-B-cell signatures" within B cells, effectively serving as a negative indicator for other cell identities. Second, feature 270 demonstrates *proxy encoding*, activating on specific ribosomal gene subgroups whose expression levels differentiate B cells from other cell types despite lacking direct biological association with B-cell function. This suggests the model exploits any discriminative pattern to construct cell representations, whether or not it aligns with canonical biological markers.

Table 1: Selected scGPT features predictive of B cell identity with key characteristics. Expr reports mean ± SD expression. # Genes denotes the number of distinct genes activated by the feature. Ribo (%) indicates the percentage of active genes that are ribosomal.

| Feat | AMI | F1 | Expr | # Genes | Ribo (%) |
|---|---|---|---|---|---|
| 478 | 0.97 | 0.99 | $18.7 \pm 6.1$ | 1485 | 0 |
| 363 | 0.95 | 0.99 | $42.6 \pm 5.8$ | 83 | 0 |
| 151 | 0.99 | 1.00 | $4.3 \pm 2.9$ | 2698 | 2 |
| 270 | 0.85 | 0.95 | $43.9 \pm 5.4$ | 187 | 85 |

**Features encode biological processes from unseen conditions.** Pre-trained scFM models can 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 training only on healthy cells, scGPT developed features that activated in disease-associated cellular states. One feature activated predominantly in monocytes and dendritic cells from patients with post-COVID-19 disorder (Figure 9), with strong enrichment for inflammation-related pathways (adjusted p-value: 1.19e-23). This aligns with clinical observations that monocytes and dendritic cells in post-COVID patients remain in persistently activated inflammatory states months after acute infection (Boes & Falter-Braun, 2023; Hopkins et al., 2023).

**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 raw gene expression values from 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 scores between its activations and the SMARTer sequencing label captured exactly this technical signature: it activated on genes with significantly greater length and lower GC content than the dataset average (Figure 3), specifically when these genes were highly expressed.

### 4.3 Features capture technical variation alongside cell type information

The CellXGene Census aggregates studies that use different protocols and sequencing technologies. A key requirement for single-cell foundation models is generalizing beyond protocol-specific artifacts to capture shared biological principles. While technical effects persist in learned represen-

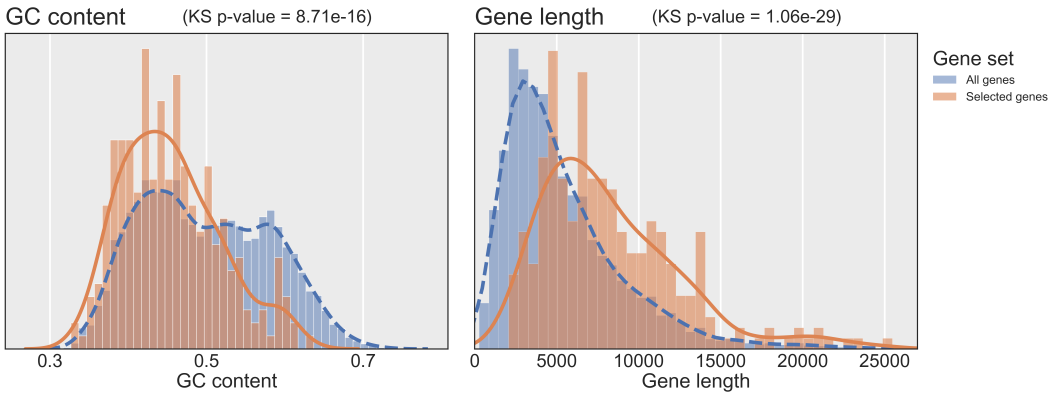


Figure 3: Density histogram of gene GC content and length distribution for all genes in the dataset (blue) and those with highest activation in feature 236 (orange).The SAE was trained on the scGPT model using the pancreas dataset. KS refers to the Kolmogorov-Smirnov test statistic.

tations unless explicitly addressed during training, robust models should identify biological concepts extending across experimental variations.

To investigate how pre-training shapes scGPT's representations, we trained a sparse autoencoder on intermediate tokens of this dataset. Cell-level association analysis revealed many features correlated with specific datasets, while others showed weaker correlations with sequencing technologies (Examples in Table 19). Though unsurprising, this demonstrates substantial inter-dataset variability and suggests that the model allocates considerable representational capacity to encoding technical differences.

Many features also showed varying generalization across studies and technologies. While some demonstrated consistent activation patterns census-wide, many associated more strongly with specific data subsets. SAE-derived cell type concepts showed variable activation, with some activating on cells from particular studies while showing reduced activity on the same cell types from other studies.

To quantify this, we compared the AMI between feature activations and complete cell type annotations versus cell type subsets spanning study subgroups. Study subsets typically showed stronger alignment than complete cell type categories (Figure 10), suggesting scGPT captures cellular identity across some studies but may not consolidate signals into unified concepts spanning all instances census-wide.

### 4.4 Cell embeddings can be integrated by deactivating specific features

Having identified that SAE features encode batch effects (Section 4.3), we tested whether these features actively contribute to technical variation in model outputs. We identified the top batch-correlated features (by AMI) in scGPT, Geneformer, and scFoundation, and clamped them to -2 during inference (see Methods 3.3). We clamped 50 features for fine-tuned models but only 20 features for pre-trained models, as batch-correlated features were less distinct in these models. Table 2 shows results for the pancreas dataset; Tables 21 and 22 (Appendix) report lung and immune results.

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. Appendix Figure 12 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.

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 (PCA) | 0.11±0.000 | 0.54±0.015 | 0.37±0.009 |
| scVI | 0.85±0.027 | 0.81±0.010 | **0.82**±0.014 |
| scFoundation (zero-shot) | 0.32±0.000 | 0.36±0.010 | 0.34±0.006 |
| scFoundation (zero-shot, steered) | 0.47±0.000 | 0.29±0.001 | **0.36**±0.006 |
| Geneformer (fine-tuned) | 0.56±0.000 | 0.86±0.007 | 0.74±0.005 |
| Geneformer (fine-tuned, steered) | 0.65±0.00 | 0.87±0.005 | **0.78**±0.003 |
| scGPT (zero-shot) | 0.60±0.016 | 0.04±0.011 | 0.26±0.010 |
| scGPT (zero-shot, steered) | 0.60±0.029 | 0.07±0.013 | **0.31**±0.014 |
| scGPT (fine-tuned) | 0.42±0.007 | 0.47±0.005 | 0.45±0.005 |
| scGPT (fine-tuned, DAR) | 0.58±0.034 | 0.58±0.007 | 0.58±0.012 |
| scGPT (fine-tuned, steered) | 0.70±0.038 | 0.56±0.005 | **0.62**±0.013 |

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.

Finally, we compared the effects of deactivating different numbers of features using steering versus randomly selected features (Figures 4 and 13). 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 15. The experiment showed that preservation of biological signal decreases with lower clamping values, while batch correction and total score reach their highest values for -2.

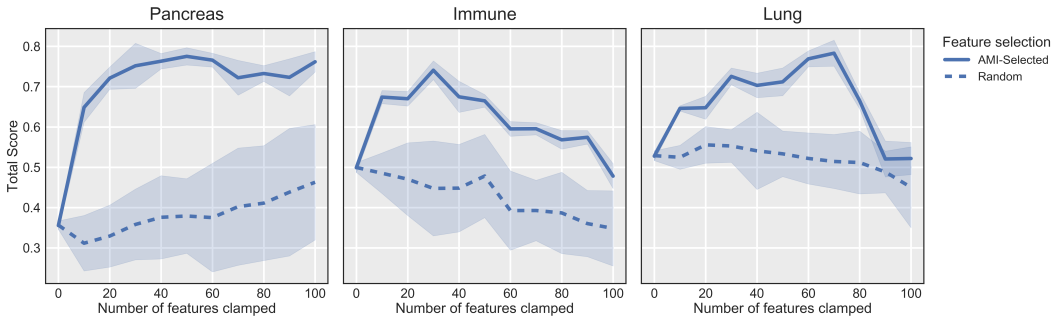


Figure 4: Total integration score as features are sequentially steered, selected randomly or by maximum AMI, across three datasets. Lines represent the mean across five seeds, and shaded regions indicate standard deviation.

## 5 Discussion & Conclusion

**scFMs learn meaningful biological representations during pre-training.** Decomposing latent representations through SAEs reveals that scFMs develop a nuanced understanding of cell biology during pre-training.The observed distinction between gene-specific and cell-specific features reflects the transformer's ability to integrate local token-level properties with global context. While gene features map directly to molecular mechanisms, cell features reveal how models synthesize distributed evidence to infer identity, often through unexpected strategies like negative encoding or

proxy markers rather than canonical biological signatures alone. 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. Standard methods such as Gene Ontology analysis prove insufficient, as many features reflect heuristic gene co-expression patterns whose biological relevance is challenging to establish. This interpretability gap is larger than in language models, where features often map to human-interpretable concepts.

**Training procedures affect information encoding.** Our cross-model comparison revealed systematic differences in how scFMs represent biological information such as gene expression encoding and cell type feature diversity. Expression encoding significantly impacts feature structure, with models adopting different strategies to deal with technology-based variance in gene expression. scGPT and Geneformer attempt to remove expression variance through binning and ranking strategies while scFoundation is explicitly trained to be robust to these fluctuations. These different strategies manifest clearly in the SAE features we found. scGPT also generated more diverse features per cell type than other models, even before fine-tuning, with notably more ribosomal-focused features. This may relate to its pre-training approach: higher masking ratios (25-75% vs. 15% in Geneformer and 30% in scFoundation) and shorter gene sequences force the model to infer cell context from broader gene sets, potentially requiring more diverse feature repertoires. These findings suggest that architectural choices can have strong effects on learned representations, warranting systematic comparison to determine which strategy better captures biological signal while minimizing technical artifacts.

**Representations reflect technical variation but exhibit limited cross-study consolidation.** scGPT dedicates substantial capacity to encoding technical differences between datasets, with features strongly correlating with specific studies or sequencing technologies. This highlights a fundamental challenge: label-free pre-training learns strong technical signals regardless of biological relevance. Variable generalization of cell type features across studies may reflect SAE limitations or genuine fragmentation in model representations. Without labels, models distinguish cells by gene expression patterns without knowing biology from technical variation. Strong technical signals from protocols, laboratory conditions, or sample preparation can overshadow shared biological patterns, causing models to learn separate representations for identical cell types across studies. Fine-tuning may be essential not for adding information, but for linking study-specific representations so models recognize separately learned patterns as instances of the same biological concept.

**SAE features are functionally related to model behavior.** Steering experiments demonstrate that high-AMI features actively encode correlated information rather than spurious associations. Suppressing batch-related features systematically improved batch integration metrics, confirming these features functionally contribute to technical variation representation and that SAEs capture functionally relevant aspects of internal representations. Steering results also show model representations are decomposable, removing specific features selectively alters behavior without disrupting other tasks. This modularity could aid understanding and controlling model behavior in single-cell applications. While current steering isn't robust enough for batch correction, it opens possibilities for sophisticated interventions. Batch information could be explicitly encoded during training for targeted removal. The ability to identify and manipulate features suggests applications in removing biases or technical artifacts from pre-trained models, analogous to concept editing in large language models.

**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, interpretability studies examining what information these models encode and how training protocols affect representations will be essential for moving toward more reliable and controllable models. The tools and findings presented here provide a foundation for such investigations.

## 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.

Constantin Ahlmann-Eltze, Florian Barkmann, Jan Lause, Valentina Boeva, and Dmitry Kobak. Representation learning of single-cell rna-seq data. *RNA*, pp. rna–080889, 2026.

Marianne Boes and Pascal Falter-Braun. Long-COVID-19: the persisting imprint of SARS-CoV-2 infections on the innate immune system. *Signal Transduct Target Ther*, 8(1):460, December 2023.

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.

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.

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.

Onkar Gujral, Mihir Bafna, Eric Alm, and Bonnie Berger. Sparse autoencoders uncover biologically interpretable features in protein language model representations. *Proceedings of the National Academy of Sciences*, 122(34):e2506316122, 2025. doi: 10.1073/pnas.2506316122. URL https://www.pnas.org/doi/abs/10.1073/pnas.2506316122.

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.

Francis R Hopkins, Melissa Govender, Cecilia Svanberg, Johan Nordgren, Hjalmar Waller, Åsa Nilsdotter-Augustinsson, Anna J Henningsson, Marie Hagbom, Johanna Sjöwall, Sofia Nyström, and Marie Larsson. Major alterations to monocyte and dendritic cell subsets lasting more than 6 months after hospitalization for COVID-19. *Front Immunol*, 13:1082912, January 2023.

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.

Samuel Marks, Adam Karvonen, and Aaron Mueller. dictionary_learning. https://github.com/saprmarks/dictionary_learning, 2024.

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.

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 (CZI Cell Science 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. 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 SCFM FINE-TUNING PROTOCOLS

In all experiments 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.
Geneformer on the other hand, was fine-tuned using supervised learning with cell type labels. However, we removed the CLS token from the latent token embeddings before analysis.

## A.4 Sparse Autoencoders

### A.4.1 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).

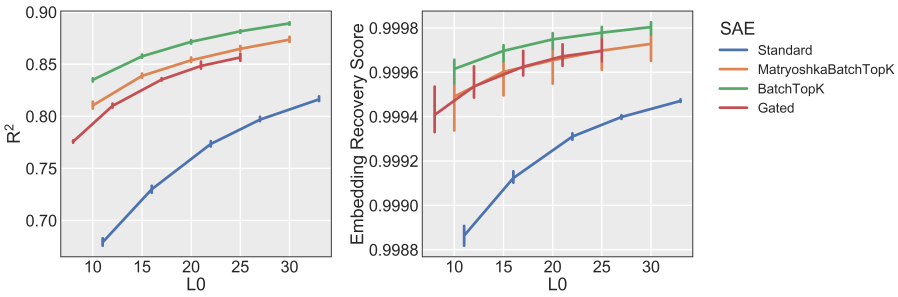


Figure 5: Performance of the four different SAE architectures at different sparsity levels trained on scGPT and the Covid-19 dataset.

### A.4.2 Embedding Recovery Score

The Embedding Recovery Score estimates the impact of token-level reconstruction on downstream embeddings by comparing cell embeddings produced by the model under different conditions:

- $\mathbf{h}_{\text{original}}$: embedding from the original input,
- $\mathbf{h}_{\text{reconstructed}}$: embedding after reconstructing the tokens,
- $\mathbf{h}_0$: ablated embedding where all tokens are zeroed out.

The score is defined as

$$\text{Embedding Recovery Score} = \frac{\text{MSE}_0 - \text{MSE}_{\text{recon}}}{\text{MSE}_0},$$

with

$$\text{MSE}_0 = \|\mathbf{h}_{\text{original}} - \mathbf{h}_0\|^2, \qquad \text{MSE}_{\text{recon}} = \|\mathbf{h}_{\text{original}} - \mathbf{h}_{\text{reconstructed}}\|^2.$$

The Embedding Recovery Score inherits some limitations from the loss recovered metric. Notably, zero-ablation has been criticized as an overly pessimistic baseline for defining the zero-point of this metric (Rajamanoharan et al., 2024). While we observed some instability in this metric during early training stages, it remained sufficiently stable for our analysis purposes.

### A.4.3 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. The SAE architecture and trainers were adapted from Marks et al. (2024).

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.4.4 Reconstruction losses

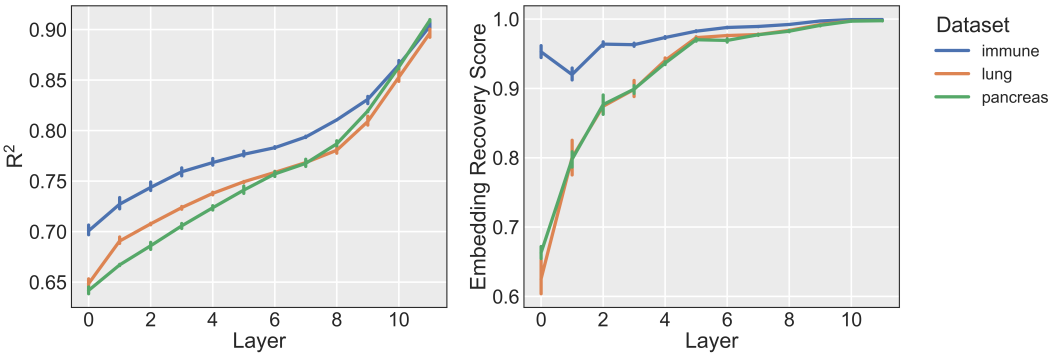


Figure 6: The final training metrics $R^2$ and Embedding Recovery Score for SAEs trained on different layers of scGPT across three datasets: Lung (blue), Immune (orange), and Pancreas (green). Lines represent the mean across three seeds, and bars indicate standard deviation.

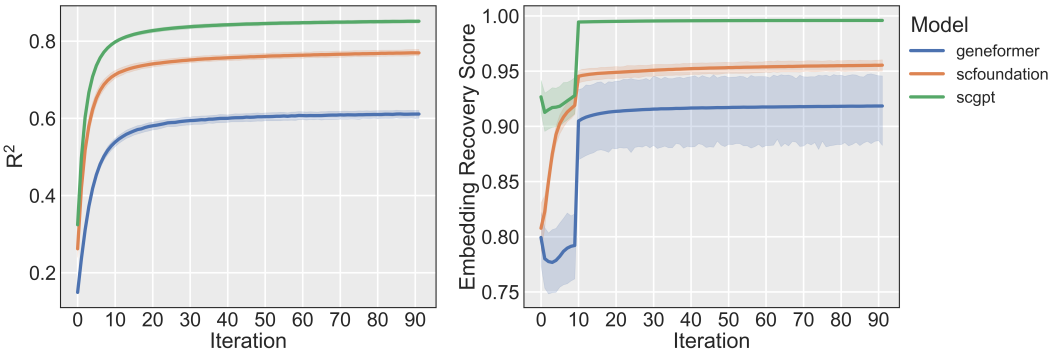


Figure 7: The training metrics $R^2$ and Embedding Recovery Score for SAEs trained on the pretrained models and the Covid-19 dataset. Lines represent the mean across the 12 layers, and shaded regions indicate standard deviation.

### A.5 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 3.2: 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.5.1 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 | $41.38 \pm 4.76$ | 0.58 | 0.18 |
| 227 | $32.16 \pm 6.37$ | 0.41 | 0.15 |
| 398 | $14.23 \pm 6.55$ | 0.44 | 0.16 |
| 92 | $5.48 \pm 4.38$ | 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 | $3.64 \pm 0.82$ | 0.38 | 0.1 |
| 451 | $1.26 \pm 0.41$ | 0.74 | 0.39 |

**Positional**
Geneformer models the relative expression of genes with their position within the sequence. A SAE trained on Geneformer will learn positional features that have a stronger activation on different parts of the sequence.

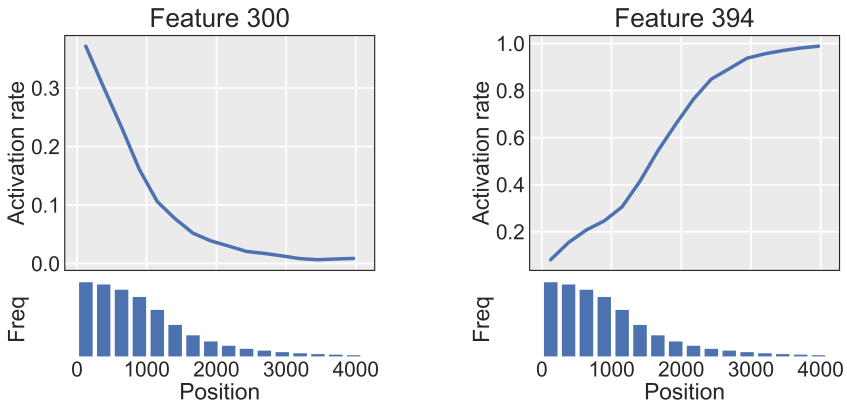


Figure 8: The activation rate of tokens at different positions in the gene sequence, normalized by the frequency a valid (non-padding) token appears at that position. Shown for features 300 and 394.

**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 |

Table 8: Gene family features in fine-tuned Geneformer (COVID-19 dataset).

| Feature | Gene family | Percentage (%) | Unique Genes |
|---|---|---|---|
| 483 | Human leukocyte antigens | 81 | 14 |
| 125 | Mitochondrially encoded protein genes | 100 | 10 |
| 38 | Histones | 82 | 67 |

**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 9: 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 10: Biological process features in pre-trained scGPT. Here the SAE was trained on the CellxGene 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 |

Table 11: Biological process features in finetuned Geneformer (COVID-19 dataset).

| Feature | Biological process | Adj. p-value |
|---|---|---|
| 279 | Cell Cycle | 5.27e-131 |
| 75 | Nuclear Outer Membrane–ER Network | 1.43e-40 |
| 201 | Programmed Cell Death | 2.59e-19 |
| 86 | Adaptive Immune Response | 7.94e-37 |
| 429 | Hemostasis | 4.10e-39 |

### A.5.2 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 12: 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 13: 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 |

Table 14: Cell type features in fine-tuned Geneformer (COVID-19 dataset)

| Feature | Cell type | AMI | F1 |
|---|---|---|---|
| 454 | T cells | 0.45 | 0.85 |
| 419 | T naive cells | 0.40 | 0.62 |
| 233 | B cells | 0.99 | 1.00 |
| 19 | B memory cells | 0.45 | 0.65 |
| 173 | Monocytes | 0.79 | 0.91 |
| 121 | Monocytes or dendritic cells | 0.86 | 0.98 |
| 156 | NK cells | 0.65 | 0.85 |
| 456 | Hematopoietic progenitor cells | 0.35 | 0.5 |
| 325 | Platelets | 1.00 | 1.00 |
| 77 | Plasma cells | 1.00 | 1.00 |

**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 15: 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 16: 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 17: 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 18: 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.5.3 SPECIFIC FEATURE EXAMPLES

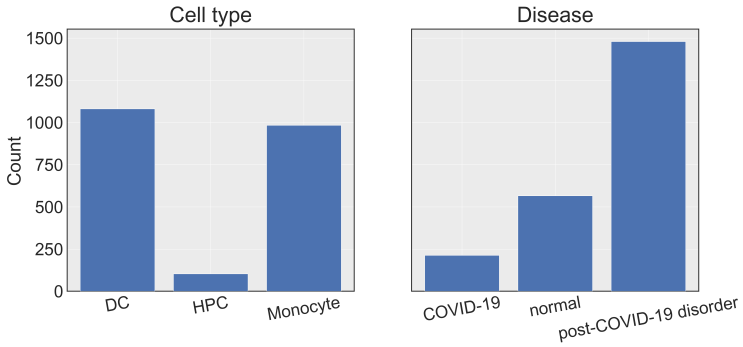


Figure 9: Distribution of token activations of featue 224 across cell types and COVID-19 status. Feature activations were thresholded above 0.5

### A.5.4 CELLXGENE BATCH EFFECTS

Table 19: Batch features for pre-trained scGPT on CellXGene Census data. Batch effects include technical variations from sequencing technologies and original studies (DS1–DS6 refer to datasets detailed in Table 20).

| Feature | Batch | AMI | F1 |
|---|---|---|---|
| 179 | DS1 | 0.63 | 0.74 |
| 337 | DS2 | 0.68 | 0.76 |
| 338 | DS3 | 0.68 | 0.76 |
| 341 | DS4 | 0.76 | 0.86 |
| 540 | DS5 | 0.92 | 0.96 |
| 832 | DS6 | 0.89 | 0.94 |
| 701 | Seq-Well | 0.31 | 0.44 |

Table 20: Characteristics of some datasets from the CellXGene Census. Note that brain tissues are overrepresented due to the composition of the CellXGene Census itself, where about half of the available datasets focus on brain-related samples.

| Dataset | Title | Tissues |
|---|---|---|
| DS1 | Infant human neocortex cells | brain |
| DS2 | Midgestational human neocortex cells | brain |
| DS3 | Retinal ganglion cells in human retina | brain |
| DS4 | Second Trimester Human Developing Brain Regions and Cortical Areas | brain |
| DS5 | L6 IT Car3 - MTG: Seattle Alzheimer's Disease Atlas (SEA-AD) | brain |
| DS6 | Human Stem Cells (LIV and TH)<br>A cell atlas of human thymic development defines T cell repertoire formation | liver, thymus |

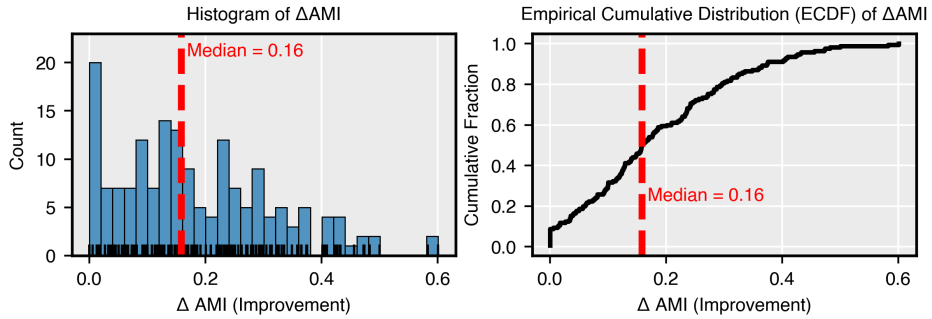


Figure 10: Distribution of the ΔAMI calculated between feature activations and two cell type annotations. Values above zero indicate features that better capture cell types within specific studies than across all studies.

### A.5.5 Cell type features over model layers

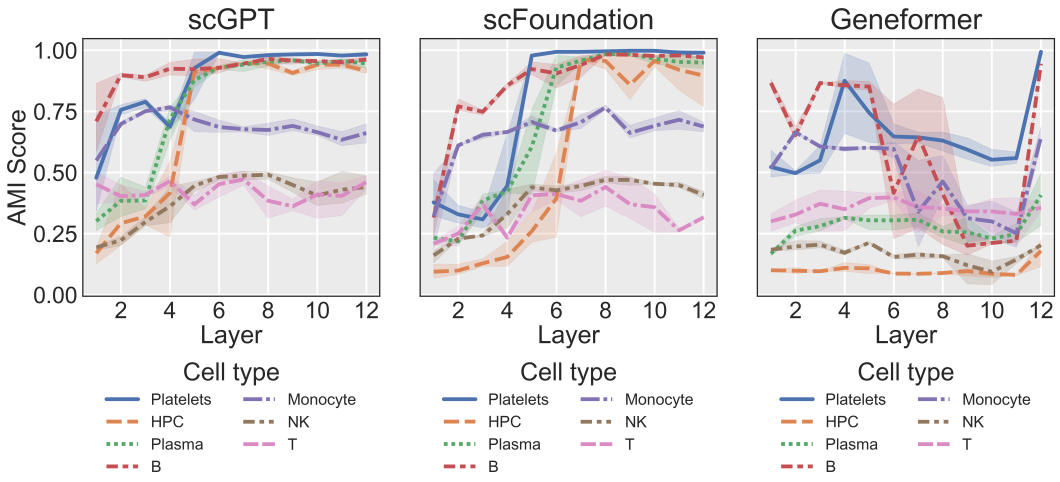


Figure 11: Layer-wise analysis of cell type feature emergence in pre-trained scGPT, scFoundation and Geneformer. For each layer, we extracted the feature most associated with each cell type by their AMI score. Each line represents a different cell type of the corresponding model. This analysis was conducted on the Covid-19 dataset.

### A.6 FEATURE STEERING

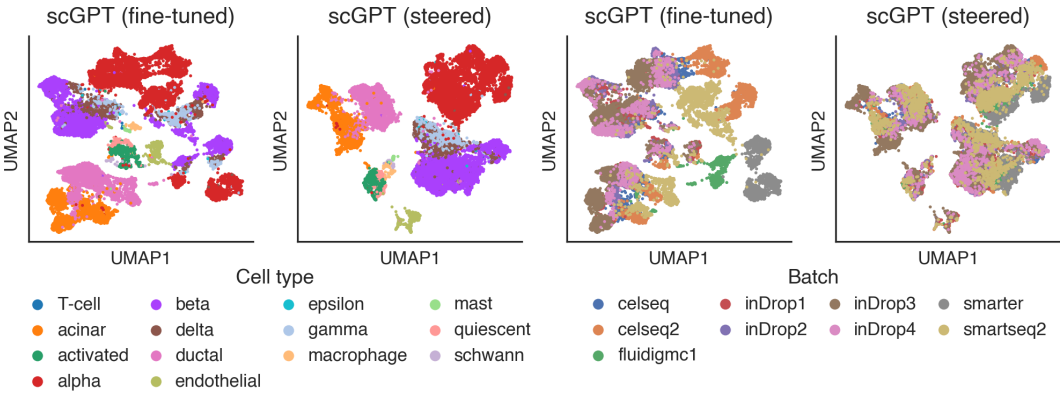


Figure 12: UMAP of the standard and steered cell embeddings of fine-tuned scGPT, colored by cell type (left) and sequencing protocol (right).

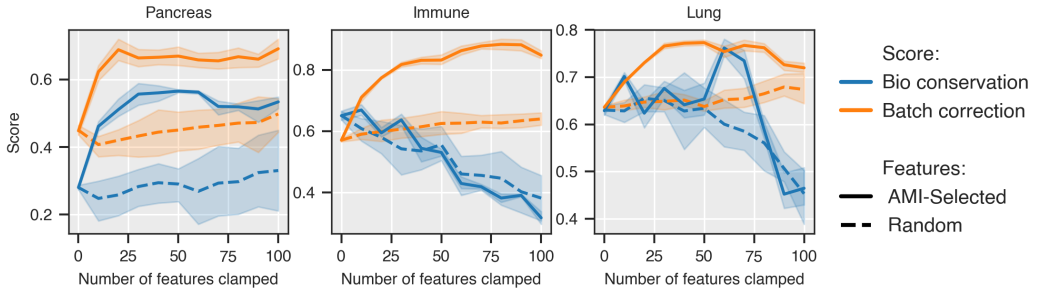


Figure 13: Biological conservation and batch correction scores as features are sequentially steered, selected randomly or by maximum AMI, across three datasets. Lines show the mean over five seeds, and shaded regions indicate standard deviation. The sparse autoencoders were trained on the fine-tuned scGPT model.

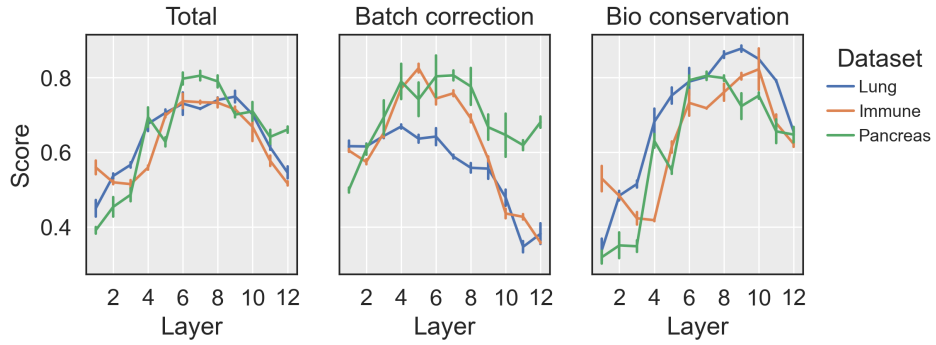


Figure 14: Layer-wise performance of SAE feature steering for batch correction. Performance metrics (total score, batch correction, and biological conservation) are shown for features extracted from different layers (1-12) of scGPT across three datasets: Lung (blue), Immune (orange), and Pancreas (green). Error bars represent standard deviation across five random seeds.

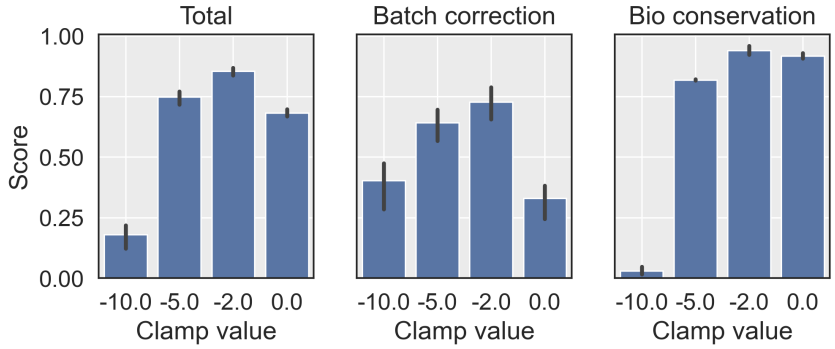


Figure 15: Effect of different clamping values on batch correction performance for scGPT. Comparison of different clamping values ($-10.0$, $-5.0$, $-2.0$, $0.0$) on three evaluation metrics: total score (left), batch correction score (middle), and biological conservation score (right). Bar heights represent mean scores across datasets, with error bars indicating standard deviation.

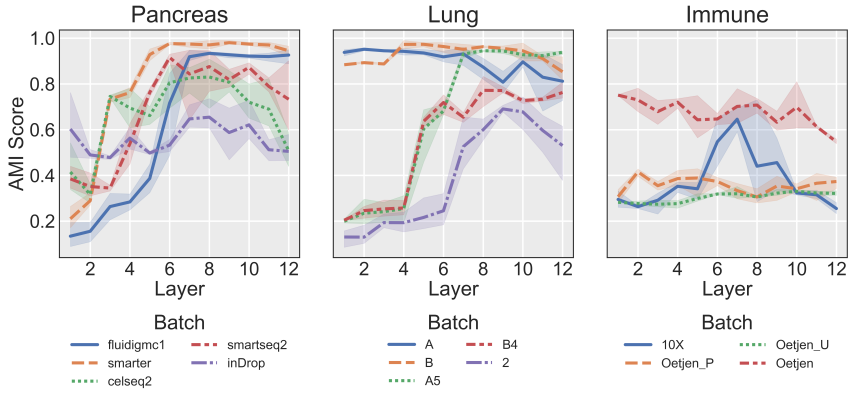


Figure 16: Layer-wise analysis of batch feature emergence in fine-tuned scGPT. For each layer, we extracted the feature most associated with each batch effect by their AMI score. Each line represents a different batch variable of the corresponding dataset.

Table 21: 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 (PCA) | 0.22±0.000 | 0.65±0.014 | 0.48±0.008 |
| scVI | 0.86±0.010 | 0.54±0.056 | **0.67**±0.003 |
| scFoundation (zero-shot) | 0.51±0.000 | 0.59±0.002 | 0.56±0.001 |
| scFoundation (zero-shot, steered) | 0.53±0.000 | 0.60±0.008 | **0.57**±0.005 |
| Geneformer (fine-tuned) | 0.49±0.000 | 0.99±0.001 | 0.79±0.001 |
| Geneformer (fine-tuned, steered) | 0.61±0.00 | 0.98±0.003 | **0.83**±0.002 |
| scGPT (zero-shot) | 0.65±0.009 | 0.08±0.003 | **0.31**±0.004 |
| scGPT (zero-shot, steered) | 0.59±0.025 | 0.01±0.002 | 0.24±0.011 |
| scGPT (fine-tuned) | 0.56±0.006 | 0.62±0.016 | 0.59±0.011 |
| scGPT (fine-tuned, DAR) | 0.68±0.003 | 0.58±0.005 | 0.62±0.003 |
| scGPT (fine-tuned, steered) | 0.71±0.003 | 0.62±0.009 | **0.65**±0.004 |

Table 22: 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 (PCA) | $0.11_{\pm 0.000}$ | $0.55_{\pm 0.004}$ | $0.37_{\pm 0.002}$ |
| scVI | $0.74_{\pm 0.008}$ | $0.73_{\pm 0.016}$ | $\mathbf{0.73}_{\pm 0.011}$ |
| scFoundation (zero-shot) | $0.47_{\pm 0.006}$ | $0.57_{\pm 0.04}$ | $\mathbf{0.53}_{\pm 0.026}$ |
| scFoundation (zero-shot, steered) | $0.44_{\pm 0.02}$ | $0.44_{\pm 0.084}$ | $0.44_{\pm 0.040}$ |
| Geneformer (fine-tuned) | $0.52_{\pm 0.000}$ | $0.91_{\pm 0.002}$ | $0.76_{\pm 0.001}$ |
| Geneformer (fine-tuned, steered) | $0.60_{\pm 0.00}$ | $0.92_{\pm 0.017}$ | $\mathbf{0.79}_{\pm 0.01}$ |
| scGPT (zero-shot) | $0.66_{\pm 0.011}$ | $0.11_{\pm 0.005}$ | $\mathbf{0.33}_{\pm 0.005}$ |
| scGPT (zero-shot, steered) | $0.78_{\pm 0.018}$ | $0.02_{\pm 0.015}$ | $0.32_{\pm 0.012}$ |
| scGPT (fine-tuned) | $0.52_{\pm 0.004}$ | $0.61_{\pm 0.010}$ | $0.58_{\pm 0.005}$ |
| scGPT (fine-tuned, DAR) | $0.75_{\pm 0.006}$ | $0.57_{\pm 0.017}$ | $0.64_{\pm 0.011}$ |
| scGPT (fine-tuned, steered) | $0.72_{\pm 0.010}$ | $0.60_{\pm 0.012}$ | $\mathbf{0.65}_{\pm 0.005}$ |

### A.7 Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

### A.8 Disclosure of LLM Editing Tools

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