LARGE-SCALE BENCHMARKING OF GENE AND EXPRESSION ENCODING STRATEGIES FOR SINGLE-CELL FOUNDATION MODELS

Igor Sadalski
igor.sadalski@gmail.com

ABSTRACT

Single-cell foundational models face a critical design choice: how to encode gene identities and expression values for transformer architectures. We present a large-scale systematic evaluation comparing learned versus pretrained protein model (ESM-2) gene embeddings, and four expression encoding strategies: discrete binning, soft binning, logarithmic binning, and continuous encoding. We train 8 model configurations from scratch on 10 million cells across 100 diverse datasets and evaluate on batch correction, biological preservation, classification, and reconstruction tasks across 26 tissue datasets. Our results demonstrate that (1) task-specific learned gene embeddings substantially outperform ESM-2 embeddings across all metrics, achieving an average 29% relative improvement; (2) soft binning achieves optimal performance with learned embeddings, achieving 91.6% classification accuracy and 86.0% macro F1 score; (3) hard binning consistently degrades performance across all evaluation metrics. The best configuration—learned embeddings with soft binning—outperforms all alternatives, providing clear guidance for single-cell model design.

1 INTRODUCTION

A major goal in recent years has been to build an AI virtual cell, i.e., multi-scale, multi-modal neural network models that can represent and simulate cellular behavior across diverse states (Bunne et al., 2024). Leading models promise to enable universal embeddings (Rosen et al., 2023), cross-species transfer (Pearce et al., 2025), multi-task transfer learning for downstream applications (Cui et al., 2024; Theodoris et al., 2023), and batch correction (Wang et al., 2022). While foundational models can be generated for different omics data types (e.g., protein, transcriptomics, tissues), here we focus on transcriptomic data. In this domain, most models (e.g., Cui et al. (2024); Adduri et al. (2025); Pearce et al. (2025)) use a transformer (Vaswani et al., 2017) backbone. Since transcriptomic data inherently consists of two pieces of information, gene identities and their expressions, model architects face the design choice of how to encode this information into embeddings that transformers can process.

Different encoding strategies embody distinct assumptions about what information is most useful. For genes, the choice is between task-specific patterns and biological priors; for expression values, it is between continuous precision and discrete robustness. For gene encoding, some models learn embeddings de novo to capture dataset-specific co-expression relationships (Cui et al., 2024), while others leverage transfer learning from pretrained protein language models such as ESM-2 (Lin et al., 2023; Adduri et al., 2025) to incorporate evolutionary biological knowledge. For expression encoding, strategies range from discretizing into bins for computational efficiency (Cui et al., 2024; Gandhi et al., 2025) to using continuous or soft binning via MLPs (Pearce et al., 2025; Ho et al., 2024; Adduri et al., 2025) or applying binning based on logarithmic transformation (Wang et al., 2022).

Most benchmarking efforts have focused on evaluating downstream applications, such as perturbation prediction (Ahlmann-Eltze et al., 2025; Wenteler et al., 2024), using already pretrained models. The most closely related work to ours is HEIMDALL (Haber et al., 2025), a modular tokenization

framework that systematically evaluated different encoding strategies but trained on only 1 million cells and did not evaluate soft binning for expression encoding.

We present a controlled benchmark that addresses these limitations by scaling pretraining tenfold to 10 million cells across 100 diverse datasets, evaluating soft binning alongside other expression encoding strategies, and performing evaluation against the Tabula Sapiens v2 benchmark. We systematically evaluate both gene and expression encoding strategies under controlled conditions with consistent model architecture and training procedures.

The main contributions of this work are:

- **A systematic evaluation of encoding strategies** comparing learned versus pretrained protein model (ESM-2) gene embeddings and four expression encoding methods (discrete binning, soft binning, logarithmic binning, and continuous encoding) under controlled conditions, including soft binning which was not evaluated by prior work.
- **A tenfold increase in pretraining scale** to 10 million cells across 100 diverse datasets, compared to 1 million cells in prior work.
- **A comprehensive evaluation** on 26 tissue datasets from Tabula Sapiens v2 spanning batch correction, biological preservation, classification, and reconstruction metrics.

2 METHODS

2.1 MODEL ARCHITECTURE AND PRETRAINING

We represented cells as bags of gene-expression pairs. Each cell i contains M_i genes with non-zero expression values, represented as a sequence of gene-expression pairs:

$$\mathcal{C}_i = ((g_{i,1}, x_{i,1}), \ldots, (g_{i,M_i}, x_{i,M_i})) \tag{1}$$

where $g_{i,j}$ denotes the j-th gene identifier and $x_{i,j}$ denotes its corresponding expression value in cell i.

Prior to encoding, we normalized expression values for each cell to a constant total c_{norm} and applied log1p transformation:

$$\tilde{x}_{i,j} = \log\left(1 + \frac{x_{i,j}}{\sum_{k=1}^{M_i} x_{i,k}} \times c_{\text{norm}}\right) \tag{2}$$

where M_i is the number of genes in cell i, c_{norm} is a normalization constant (typically 10^4 or 10^6), and k indexes over all genes in cell i. This normalization step ensures consistent scaling across cells with varying sequencing depths.

Because transformers scale with the square of the size of the context window (number of inputs), we employed a sampling strategy to select a subset of genes for each cell. Let $S : \mathcal{C}_i \to \{(g_{i,j}, x_{i,j})\}_{j=1}^{K}$ be a sampling function that takes cell i and returns K gene-expression pairs, where $K = \min(\text{context\_window}, M_i)$. During training, for datasets where the number of non-zero expressed genes exceeded the context window, we randomly sampled K genes from all non-zero expressed genes in each cell:

$$\mathcal{S}_{\text{random}}^{(i)} = \text{RandomSample}(\{g_j : x_{i,j} > 0\}, K) \tag{3}$$

This approach ensures diverse gene representation across training examples while maintaining computational feasibility for large gene vocabularies.

The training objective was to predict masked expression values given the gene identities and unmasked expression context. To this end, we first computed embeddings for both gene identities and expression values. The gene embedding for each gene was obtained as:

$$\boldsymbol{e}_g^{(i,j)} = \text{enc}_g(g_{i,j}) \in \mathbb{R}^{d_g}, \tag{4}$$

where enc_g is the gene encoding function (detailed in Section 2.2), d_g is the gene embedding dimension, and $\boldsymbol{e}_g^{(i,j)} \in \mathbb{R}^{d_g}$ is the gene embedding vector. For expression values, we randomly masked

a fraction p_{mask} of gene-expression pairs during training. The expression embedding was computed as:

$$\boldsymbol{e}_x^{(i,j)} = \begin{cases} \text{enc}_x(\tilde{x}_{i,j}) & \text{with probability } 1 - p_{\text{mask}} \\ \boldsymbol{m} & \text{with probability } p_{\text{mask}} \end{cases} \tag{5}$$

where enc_x is the expression encoding function (detailed in Section 2.3), $\boldsymbol{e}_x^{(i,j)} \in \mathbb{R}^{d_x}$ is the expression embedding vector, p_{mask} is the masking probability, and $\boldsymbol{m} \in \mathbb{R}^{d_x}$ is a learnable mask token with dimension d_x matching the expression embedding dimension.

The combined gene-expression embedding for the j-th gene in cell i was obtained by summing the gene and expression embeddings:

$$\boldsymbol{z}_0^{(i,j)} = \boldsymbol{e}_g^{(i,j)} + \boldsymbol{e}_x^{(i,j)} \tag{6}$$

where $\boldsymbol{z}_0^{(i,j)} \in \mathbb{R}^d$ is the combined embedding for gene j in cell i, and d is the model dimension (equal to both d_g and d_x). The combined embeddings for all K genes in cell i were concatenated together to form the input sequence for the model:

$$\boldsymbol{z}_0^{(i)} = [\boldsymbol{z}_0^{(i,1)}, \ldots, \boldsymbol{z}_0^{(i,K)}] \tag{7}$$

where $\boldsymbol{z}_0^{(i)} \in \mathbb{R}^{K \times d}$ is the input sequence for cell i containing K gene embeddings. These combined embeddings were then passed through a transformer encoder. Grouping the gene embeddings per cell and recursively applying the transformer layers, we obtained:

$$\boldsymbol{z}_l^{(i)} = f_{\text{transformer}}(\boldsymbol{z}_{l-1}^{(i)}) \tag{8}$$

where $f_{\text{transformer}}$ denotes a standard transformer encoder layer, $l \in \{1, \ldots, n\}$ indexes the layers, and n is the total number of transformer layers.

The predicted expression value for the j-th gene in cell i was decoded from the final transformer layer output using a simple MLP:

$$\bar{x}_{i,j} = \text{MLP}(\boldsymbol{z}_n^{(i,j)}) \tag{9}$$

where $\boldsymbol{z}_n^{(i,j)} \in \mathbb{R}^d$ is the output embedding for gene j in cell i from the final transformer layer, and $\bar{x}_{i,j} \in \mathbb{R}$ is the predicted expression value. We optimized the model using mean squared error loss for reconstruction of masked gene expression values (Wang et al., 2022; Adduri et al., 2025; Ho et al., 2024; Cui et al., 2024):

$$\mathcal{L}_{i,j} = \frac{1}{|\mathcal{U}_{\text{unk}}|} \sum_{j \in \mathcal{U}_{\text{unk}}} (\tilde{x}_{i,j} - \bar{x}_{i,j})^2 \tag{10}$$

where $\mathcal{L}_{i,j}$ is the loss for gene j in cell i, $\mathcal{U}_{\text{unk}}$ represents the set of masked gene indices, $\tilde{x}_{i,j}$ is the normalized true expression value, and $\bar{x}_{i,j}$ is the predicted expression value from Equation 9.

2.2 Gene Encoding Strategies

Learned encoding is obtained by passing the gene identifier through a learned embedding table (Cui et al., 2024; Gandhi et al., 2025). This approach allows the model to generate task-specific representations for each gene, enabling it to capture dataset-specific relationships.

$$\text{enc}_g^{\text{learned}}(g_{i,j}) = \text{Embedding}(g_{i,j}) \tag{11}$$

ESM-2 encoding uses precomputed embeddings retrieved from the ESM-2 (3B) model dictionary (Lin et al., 2023). STATE (Adduri et al., 2025) leverages this approach, using large pretrained protein language models to provide biologically-informed representations that are consistent across datasets and species, making it a strong choice for transfer learning and handling new or rare gene symbols. We project these embeddings through an MLP to match the model dimension.

$$\text{enc}_g^{\text{ESM-2}}(g_{i,j}) = \text{MLP}(\mathcal{E}(g_{i,j})) \tag{12}$$

where $\mathcal{E}(g_{i,j})$ denotes the precomputed ESM-2 embedding for gene identifier $g_{i,j}$, and MLP is a multi-layer perceptron that projects the embedding to the model dimension.

2.3 EXPRESSION ENCODING STRATEGIES

Raw expression encoding uses a simple MLP on normalized data. This approach preserves the full, continuous information from the expression measurement and is the most direct way to encode quantitative gene expression levels.

$$\text{enc}_x^{\text{raw}}(\tilde{x}_{i,j}) = \text{MLP}(\tilde{x}_{i,j}) \tag{13}$$

Hard binning encoding discretizes expression values into bins. This method groups expression levels into discrete intervals, trading off resolution for robustness and simplifying the input space (Cui et al., 2024; Gandhi et al., 2025).

$$b_{i,j} = \begin{cases} k, & \text{if } x_{i,j} > 0 \text{ and } x_{i,j} \in [\beta_k, \beta_{k+1}], \\ 0, & \text{if } x_{i,j} = 0, \end{cases} \tag{14}$$

where $b_{i,j}$ is the bin index for the expression value of gene j in cell i, k is the bin index, and β_k and β_{k+1} are the lower and upper boundaries of bin k, respectively. Expression embedding is obtained using an embedding layer:

$$\text{enc}_x^{\text{hard bin}}(x_{i,j}) = \text{Embedding}(b_{i,j}) \tag{15}$$

Log binning encoding compresses a wide dynamic range of expression values using a logarithmic transformation before discretizing. This approach can potentially mitigate the effects of outliers or skewed distributions. scBERT (Wang et al., 2022) employs log binning with discrete tokenization.

$$b_{i,j} = \min\left(\left\lfloor \log_2\left(x_{i,j} + 1\right)\right\rfloor, B_{\max}\right) \tag{16}$$

where $b_{i,j}$ is the bin index for the expression value of gene j in cell i, and $B_{\max}$ is the maximum bin index. Bins are embedded using an embedding layer:

$$\text{enc}_x^{\text{log bin}}(x_{i,j}) = \text{Embedding}(b_{i,j}) \tag{17}$$

Soft binning encoding uses a softmax over potential bins to allow fractional/bin-weighted expression, which can capture uncertainty and subtle intensity differences between expression values, making the encoding differentiable and potentially more expressive (Ho et al., 2024; Hao et al., 2024; Pearce et al., 2025).

$$\boldsymbol{\alpha}_x^{(i,j)} = \text{Softmax}\left(\boldsymbol{W}_{x,2}\text{LeakyReLU}\left(\boldsymbol{W}_{x,1}\, x_{i,j}\right)\right) \tag{18}$$

where $\boldsymbol{\alpha}_x^{(i,j)}$ is a vector of bin weights for the expression value of gene j in cell i, $\boldsymbol{W}_{x,1}$ and $\boldsymbol{W}_{x,2}$ are learnable weight matrices, and LeakyReLU is the LeakyReLU activation function. The expression embedding is obtained by performing a soft lookup in the embedding table:

$$\text{enc}_x^{\text{soft bin}}(x_{i,j}) = \sum_{k=1}^{b} \alpha_{x,k}^{(i,j)} \boldsymbol{T}_k \tag{19}$$

where b is the number of bins, $\alpha_{x,k}^{(i,j)}$ is the k-th element of $\boldsymbol{\alpha}_x^{(i,j)}$, and $\boldsymbol{T}_k$ is the k-th embedding vector in the embedding table $\boldsymbol{T}$.

3 EXPERIMENTS

3.1 TRAINING

We trained our models on 101 diverse single-cell RNA sequencing datasets comprising over 10 million cells, randomly selected from the curated collection used to train the Transcriptformer model (Pearce et al., 2025). The datasets span diverse tissue types, experimental protocols, species, biological conditions, developmental stages, and disease states (see Appendix A). We evaluated four different expression encoding strategies and two different gene encoding strategies by training models for all combinations of these strategies (8 models total). During the training process, we used the hyperparameters defined in Table 1 in the Appendix, which were based on previous works (Cui et al., 2024; Adduri et al., 2025; Pearce et al., 2025). All models were trained using automatic mixed precision (AMP) with FP16 to accelerate training. During training, we recorded the reconstruction loss for each model using Mean Squared Error on masked gene expression values. We also recorded the best model checkpoints based on the validation reconstruction loss.

3.2 EVALUATION

We evaluated all models on 26 tissue-specific datasets from the Tabula Sapiens v2 benchmark (Tabula Sapiens Consortium et al., 2022), comprising more than half a million cells across diverse human tissues. This benchmark represents a standard evaluation protocol in the field and has been used to evaluate other foundational models (Pearce et al., 2025). Using each trained model, we embedded the 26 evaluation datasets. For each embedded evaluation dataset, we generated cell embeddings by extracting token embeddings from the final encoder layer (where each token represents a gene-expression pair) and mean-pooling these embeddings across all valid (non-padded) positions for each cell. We annotated our embeddings with cell type. Additionally, for the batch key, we used the `10X_run` identifier, which distinguishes cells from different 10X Genomics sequencing runs.

Batch correction metrics. To assess batch effect removal, we used the scib package (Luecken et al., 2022) and computed: Integration Local Inverse Simpson's Index, which measures the diversity of batch labels in the local neighborhood of each cell to quantify batch mixing; Cell-type Local Inverse Simpson's Index, which evaluates the preservation of biological structure by measuring the diversity of cell-type labels in the local neighborhood; Average Silhouette Width for batch, which measures how well batches are mixed globally using average silhouette width; and graph connectivity, which assesses whether cells sharing the same label form a fully connected subgraph in the k-nearest neighbors graph.

Biological preservation metrics. Similar to batch correction metrics (also using the scib package (Luecken et al., 2022)), we computed: Average Silhouette Width for label, which quantifies how well cells of the same cell type cluster together using average silhouette width computed over cell-type labels; Normalized Mutual Information, which measures the normalized mutual information between predicted clusters and true cell-type labels; Adjusted Rand Index, which computes the adjusted rand index between predicted clusters and true cell-type labels; isolated label F1 score, which evaluates the F1 score for cell types that are isolated in the embedding space; and isolated label silhouette, which measures the silhouette score for isolated cell types.

Cell type classification. To compute cell type classification performance, we extracted cell embeddings by mean-pooling token embeddings from the final encoder layer (where each token represents a gene-expression pair) (Pearce et al., 2025). We filtered out cells with missing cell type annotations and removed cell types with fewer than 250 cells to ensure robust evaluation. We split the data into training and test sets using an 80/20 stratified split. We trained a k-nearest neighbors classifier with $k = 10$ on the training set and evaluated performance on the held-out test set.

4 RESULTS

We evaluated all 8 model configurations (2 gene encoding strategies $\times$ 4 expression encoding strategies) on 26 tissue-specific datasets from the Tabula Sapiens v2 benchmark. For each encoder configuration, we embedded each of the 26 datasets separately and then computed selected metrics from the scib package (Luecken et al., 2022) for evaluating batch correction and biological preservation, as well as cell type classification performance and pretraining reconstruction loss. Our evaluation revealed three key findings: (1) task-specific learned gene embeddings substantially outperform ESM-2 embeddings across all metrics, (2) soft binning with learned embeddings emerged as the optimal architecture, and (3) the commonly used hard binning expression encoding strategy Cui et al. (2024); Gandhi et al. (2025) degrades model performance across all evaluation metrics.

4.1 BATCH CORRECTION AND BIOLOGICAL PRESERVATION

Figure 1 shows a comparison of models with different gene and expression encoders and their ability to perform batch correction and preserve biological features when generating embeddings on a new dataset. We obtained broadly similar results to other studies reporting on these metrics (Luecken et al., 2022) (though in our setup, we used only a subset of the metrics used in the original study). Focusing on expression encoding strategies, learned soft binning yields the best overall performance. Compared to learned hard binning, soft binning provides a 79.2% relative improvement in biological preservation and a 33.0% improvement in batch correction. When compared to the second-best method (learned raw encoding), soft binning achieves a 16.0% relative improvement

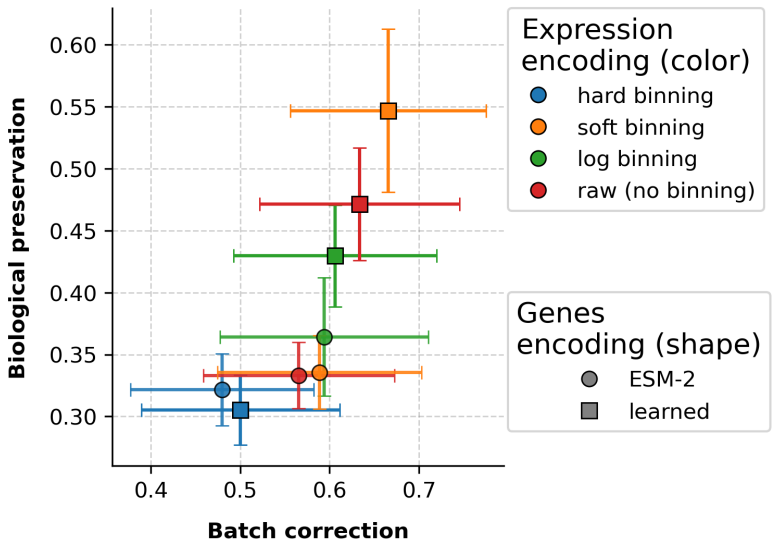

Figure 1: Performance comparison across encoding strategies on batch correction and biological preservation metrics. We evaluated each model configuration on 11 of 26 evaluation datasets (across the most varied tissues). For each dataset, we first computed multiple evaluation metrics, then grouped them into two meta-metrics: batch correction and biological preservation. Within each group, we averaged the metrics to obtain per-dataset means for each model configuration. We then aggregated results across datasets by computing the mean and standard deviation of these per-dataset means (error bars represent standard deviations across datasets). The x-axis shows batch correction performance, computed as the average of: Integration Local Inverse Simpson's Index, graph connectivity, and Average Silhouette Width for batch. The y-axis shows biological preservation performance, computed as the average of: Average Silhouette Width for label, Cell-type Local Inverse Simpson's Index, Normalized Mutual Information, Adjusted Rand Index, isolated label F1 score, and isolated label silhouette score. Colors indicate expression encoding strategies (hard binning, soft binning, log binning, raw expression); marker shapes indicate gene encoding strategies (learned embedding or ESM-2 embedding).

in biological preservation and a 5.1% improvement in batch correction. For gene encoding strategies, learned gene embeddings provide a 29% relative improvement over ESM-2 embeddings on biological preservation and 8% on batch correction.

4.2 Cell Type Classification Performance

Figure 2 shows cell type classification performance across all encoding configurations. Learned gene embeddings outperformed ESM-2 embeddings in three out of four cases, achieving 31% higher accuracy and 58% higher macro F1 score on average. The best-performing configuration combines learned gene embeddings with soft binning, achieving 91.6% accuracy and 86.0% macro F1 score. This represents a 92% improvement in macro F1 over hard binning with learned embeddings (44.8% macro F1). The order of performance for different methods, in both accuracy and macro F1, is consistent with the other results (Figures 1 and 3).

4.3 Pretraining Reconstruction Quality

Figure 3 shows the minimum training reconstruction loss achieved during pretraining for each model configuration. Configurations with learned gene embeddings achieved 21% lower minimum training loss than ESM-2 configurations (0.046 vs. 0.058 MSE on average, where lower is better). These results align with the other evaluation metrics. This correlation suggests that better pretraining reconstruction quality translates to improved downstream task performance. The higher training loss for ESM-2 configurations, combined with observed training instability (loss divergence in all

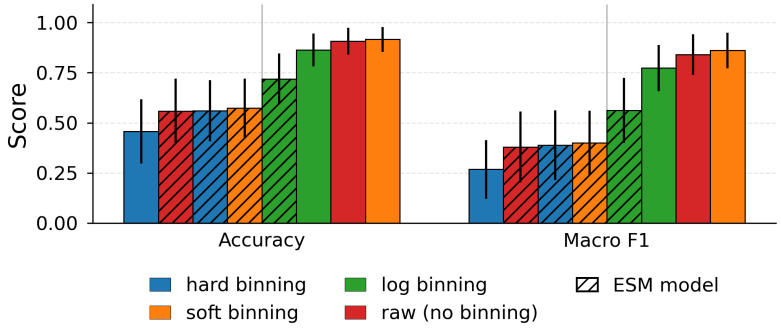

Figure 2: Cell type classification performance across encoding strategies. To obtain each mean and standard deviation for accuracy and macro F1 score, we trained a kNN classifier on the embeddings from the trained model (Section 3.2) for one of the evaluation datasets. We then selected one of the pretrained models (with different architecture and encoding strategies). We generated embeddings for the evaluation dataset using that model. We then created a subset of the embeddings for that dataset and computed several evaluation metrics, then averaged those metrics within the dataset to obtain a per-dataset mean for each model configuration. We aggregated results by taking the mean and standard deviation of these per-dataset metric means across all datasets (error bars represent standard deviations of the per-dataset means). Bars show mean accuracy and macro F1 score with error bars indicating standard deviation across 26 evaluation datasets. Colors indicate expression encoding strategies (hard, soft, log, raw); hatching (///) indicates ESM-2 gene encoding. Configurations are sorted in ascending order.

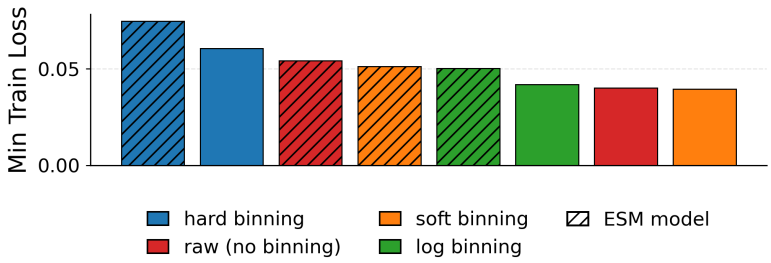

Figure 3: Minimum Mean Squared Error (MSE) training loss achieved during pretraining for each model configuration. Bars are sorted by minimum loss in descending order. Colors indicate expression encoding strategies (hard, soft, log, raw); hatching (///) indicates ESM-2 gene encoding.

four ESM-2 cases and learned hard binning), indicates that transfer learning from protein language models may introduce optimization challenges that limit effectiveness for transcriptomic tasks.

5 Discussion

Learned gene embeddings outperform ESM-2 encoding. Despite using an MLP layer to project ESM-2 embeddings to the model dimension, models trained with ESM-2 embeddings showed unstable training with loss divergence in all four cases. Learned embeddings achieved an average 29% relative improvement over ESM-2 embeddings across all metrics (29% improvement in biological preservation, 8% in batch correction, 31% higher accuracy, 58% higher macro F1, 21% lower pretraining reconstruction loss). This gap suggests that task-specific representations capture dataset-specific co-expression relationships more relevant for transcriptomic tasks than evolutionary protein-level information. Additionally, single-cell data includes many non-coding genes, pseudogenes, and gene isoforms not well-represented in protein language models.

Soft binning provides optimal expression encoding. Soft binning balances discrete and continuous representations. When combined with learned gene embeddings, it achieves the highest scores across all the metrics tested here. We hypothesize that this is because, unlike hard binning, soft binning preserves information at bin boundaries and captures subtle expression differences, while providing regularization that may help learn more robust representations than raw continuous encoding.

Our evaluation is limited to human tissues and a single shared model architecture and training procedure. Future work should evaluate additional downstream tasks and species, investigate alternative architectures, design new gene encoding methods, and provide mechanistic insights into why learned embeddings and soft binning outperform their alternatives.

6 Conclusion

We present a large-scale systematic benchmark comparing gene and expression encoding strategies, training 8 model configurations on 10 million cells across 100 diverse datasets under controlled conditions. Our evaluation across 26 tissue datasets from the Tabula Sapiens v2 benchmark yields three key findings.

First, task-specific learned gene embeddings substantially outperform pretrained protein model (ESM-2) embeddings across all metrics, achieving an average 29% relative improvement. Second, the best-performing configuration—learned gene embeddings with soft binning—achieves superior performance across all evaluation dimensions: 91.6% classification accuracy and 86.0% macro F1 score, representing a 92% improvement over hard binning. Third, commonly used hard binning expression encoding strategies degrade model performance across all metrics. These results, obtained at 10x larger scale than prior work (Haber et al., 2025), provide clear empirical guidance for single-cell foundation model design: learned gene embeddings with soft binning should be the default choice for within-species transcriptomic tasks.

Impact Statement

This paper presents a benchmark study that provides empirical guidance for designing effective foundational models in single-cell RNA sequencing analysis. Potential positive impacts include enabling more effective analysis tools for biomedical research and providing reproducible benchmarks for method comparison. Potential negative impacts are limited, as this is a methodological study focused on model architecture rather than direct applications.

Acknowledgments

We thank Matthew Bernstein, Laura Bagamery, and Dan Raviv for their help and guidance throughout this work.

References

Abhinav K Adduri, Dhruv Gautam, Beatrice Bevilacqua, Alishba Imran, Rohan Shah, Mohsen Naghipourfar, Noam Teyssier, Rajesh Ilango, Sanjay Nagaraj, Mingze Dong, Chiara Ricci-Tam, Christopher Carpenter, Vishvak Subramanyam, Aidan Winters, Sravya Tirukkovalur, Jeremy Sullivan, Brian S Plosky, Basak Eraslan, Nicholas D Youngblut, Jure Leskovec, Luke A Gilbert, Silvana Konermann, Patrick D Hsu, Alexander Dobin, Dave P Burke, Hani Goodarzi, and Yusuf H Roohani. Predicting cellular responses to perturbation across diverse contexts with state. *bioRxiv*, 2025. doi: 10.1101/2025.06.26.661135. URL https://doi.org/10.1101/2025.06.26.661135. Preprint.

Constantin Ahlmann-Eltze, Wolfgang Huber, and Simon Anders. Deep-learning-based gene perturbation effect prediction does not yet outperform simple linear baselines. *Nature Methods*, 22: 1657–1661, 2025. doi: 10.1038/s41592-025-02772-6. URL https://doi.org/10.1038/s41592-025-02772-6.

Charlotte Bunne, Yusuf Roohani, Yanay Rosen, others, Emma Lundberg, Jure Leskovec, and Stephen R Quake. How to build the virtual cell with artificial intelligence: Priorities and opportunities. *Cell*, 187(25), 2024.

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*, 21:1470–1480, 2024. doi: 10.1038/s41592-024-02201-0. URL https://doi.org/10.1038/s41592-024-02201-0.

Shreshth Gandhi, Farnoosh Javadi, Valentine Svensson, Umair Khan, Matthew G. Jones, John Yu, Daniele Merico, Hani Goodarzi, and Nima Alidoust. Tahoe-x1: Scaling perturbation-trained single-cell foundation models to 3 billion parameters. *bioRxiv*, 2025. doi: 10.1101/2025.10.23.683759. URL https://doi.org/10.1101/2025.10.23.683759. Preprint.

Ellie Haber, Shahul Alam, Nicholas Ho, Renming Liu, Evan Trop, Shaoheng Liang, Muyu Yang, Spencer Krieger, and Jian Ma. Heimdall: A modular framework for tokenization in single-cell foundation models. *bioRxiv*, 2025. doi: 10.1101/2025.11.09.687403. URL https://doi.org/10.1101/2025.11.09.687403. Preprint.

M. Hao, J. Gong, X. Zeng, et al. Large-scale foundation model on single-cell transcriptomics. *Nature Methods*, 21:1481–1491, 2024. doi: 10.1038/s41592-024-02305-7. URL https://doi.org/10.1038/s41592-024-02305-7.

Nicholas Ho, Caleb N Ellington, Jinyu Hou, Sohan Addagudi, Shentong Mo, Tianhua Tao, Dian Li, Yonghao Zhuang, Hongyi Wang, Xingyi Cheng, Le Song, and Eric P Xing. Scaling dense representations for single cell with transcriptome-scale context. *bioRxiv*, 2024. doi: 10.1101/2024.11.28.625303. URL https://doi.org/10.1101/2024.11.28.625303. Preprint.

Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaron Shmueli, Allan dos Santos Costa, Maryam Fazel-Zarandi, Tom Sercu, Salvatore Candido, and Alexander Rives. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023. doi: 10.1126/science.ade2574. URL https://doi.org/10.1126/science.ade2574.

Malte D. Luecken, M. Büttner, K. Chaichoompu, A. Danese, M. Interlandi, M. F. Mueller, D. C. Strobl, L. Zappia, M. Dugas, M. Colomé-Tatché, and Fabian J. Theis. Benchmarking atlas-level data integration in single-cell genomics. *Nature Methods*, 19:41–50, 2022. doi: 10.1038/s41592-021-01336-8. URL https://doi.org/10.1038/s41592-021-01336-8.

James D Pearce, Sara E Simmonds, Gita Mahmoudabadi, Lakshmi Krishnan, Giovanni Palla, Ana-Maria Istrate, Alexander Tarashansky, Benjamin Nelson, Omar Valenzuela, Donghui Li, Stephen R Quake, and Theofanis Karaletsos. A cross-species generative cell atlas across 1.5 billion years of evolution: The transcriptformer single-cell model. *bioRxiv*, 2025. doi: 10.1101/2025.04.25.650731. URL https://doi.org/10.1101/2025.04.25.650731. Preprint.

Yanay Rosen, Yusuf Roohani, Ayush Agrawal, Leon Samotorčan, Tabula Sapiens Consortium, Stephen R Quake, and Jure Leskovec. Universal cell embeddings: A foundation model for cell biology. *bioRxiv*, 2023. doi: 10.1101/2023.11.28.568918. URL https://doi.org/10.1101/2023.11.28.568918. Preprint.

Tabula Sapiens Consortium, Robert C Jones, Jim Karkanias, Mark A Krasnow, Angela Oliveira Pisco, Stephen R Quake, Julia Salzman, Nir Yosef, Bryan Bulthaup, Phillip Brown, Will Harper, Michael Hemenez, Ravikumar Ponnusamy, Ahmad Salehi, Bhavani A Sanagavarapu, Eileen Spallino, Kalleen A Aaron, Waldo Concepcion, James M Gardner, Brendan Kelly, Nicholas Neidlinger, Zifa Wang, Sheela Crasta, Saroja Kolluru, Maurizio Morri, Yan Tan, Kyle J Travaglini, Chenling Xu, Maria Alimova, Nicholas E Banovich, Ben A Barres, Philip A Beachy, Biter Bilen, Douglas Brownfield, Charles K F Chan, Songming Chen, Michael F Clarke, Sabrina D Conley, Spyros Darmanis, Aaron Demers, Kubilay Demir, Antoine de Morree, Tony Divita, Haley du Bois, Hamid Ebadi, F Hernan Espinoza, Matt Fish, Qiang Gan, Benson M George, Jeffrey M Granja, Foad Green, Gunsagar S Gulati, Michael S Haney, Julie A Harris, Yanzhe He, Shayan Hosseinzadeh, Albin Huang, Kerwyn Casey Huang, Atsushi Iriki, Eric Jean, Kevin S

Kao, Guruswamy Karnam, Aaron M Kershner, Bernhard M Kiss, William Kong, Maya E Kumar, Jonathan Lam, Song E Lee, Benoit Lehallier, Qiang Li, Yan Li, Ling Liu, Annie Lo, Wan-Jin Lu, Marisol F Lugo-Fagundo, Anjali Manjunath, Andrew P May, Ashley Maynard, Aaron McGeever, Madeleine McKay, Michael I Miller, Mais Moussa, Ravi Mylvaganam, EK Neumann, Joseph Noh, Roel Nusse, Irene Papatheodorou, Traci Peng, Lolita Penland, Katherine Pollard, Robert Puccinelli, Zhen Qi, Stephen R Quake, Thomas A Rando, Micha Sam Raredon, Karine Rizzoti, Katherine Rogers, Yanay Rosen, M Elizabeth Rothenberg, Meritxell Rovira, Yaroslava Ruzankina, Nicholas Schaum, Eran Segal, Jun Seita, Rahul Sinha, Rene V Sit, Justin Sonnenburg, Christof Stringer, Kai Tan, Michelle Tan, Sudhir Gopal Tattikota, Kyle J Travaglini, Carolina Tropini, Michelle Tsui, Lucas Waldburger, Bruce M Wang, Linda J van Weele, Brice M Weinstein, Michael N Wosczyna, Angela Wu, Jinyi Xiang, Sizun Xue, Kevin A Yamauchi, Andrew C Yang, Lakshmi P Yerra, Justin Youngyunpipatkul, Bo Yu, Fabio Zanini, Gizem Zardeneta, Tiffany Zee, Chunyu Zhao, Fan Zhang, Hui Zhang, Martin Jinye Zhang, Lu Zhou, and Daniel R Zollinger. The tabula sapiens: A multiple-organ, single-cell transcriptomic atlas of humans. *Science*, 376(6594):eabl4896, 2022. doi: 10.1126/science.abl4896. URL https://doi.org/10.1126/science.abl4896.

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, and Patrick T Ellinor. Transfer learning enables predictions in network biology. *Nature*, 618(7965):616–624, 2023. doi: 10.1038/s41586-023-06139-9. URL https://doi.org/10.1038/s41586-023-06139-9.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 2017. URL https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Wenchuan Wang, Fan Yang, 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, 2022. doi: 10.1038/s42256-022-00534-z. URL https://doi.org/10.1038/s42256-022-00534-z.

A. Wenteler, M. Occhetta, N. Branson, M. Huebner, V. Curean, W. T. Dee, W. T. Connell, A. Hawkins-Hooker, S. P. Chung, Y. Ektefaie, A. Gallagher-Syed, and C. M. V. Córdova. Pertevalscfm: Benchmarking single-cell foundation models for perturbation effect prediction. *bioRxiv*, 2024. doi: 10.1101/2024.10.02.616248. URL https://doi.org/10.1101/2024.10.02.616248. Preprint.

A Additional Details

A.1 Reproducibility

Training datasets are from publicly available sources (Pearce et al., 2025). Evaluation datasets from Tabula Sapiens v2 are available from the official repository (Tabula Sapiens Consortium et al., 2022).

A.2 Hyperparameters

Table 1 summarizes the hyperparameters used for training all models in this study. These hyperparameters were based on previous works (Cui et al., 2024; Adduri et al., 2025; Pearce et al., 2025).

A.3 Training Datasets

Table 2 lists all 101 training datasets used in this study, including the dataset identifier and number of cells for each dataset. The total number of cells across all training datasets is 10,010,835.

Table 1: Hyperparameters used for model training (in part adopted from open source code from various works (Cui et al., 2024; Adduri et al., 2025; Pearce et al., 2025; Gandhi et al., 2025)).

Hyperparameter	**Value**
Model Architecture	
Model dimension (d_{model})	512
Number of transformer blocks (n_{blocks})	12
Number of attention heads (n_{head})	8
Feed-forward dimension (d_{hid})	1024
Dropout	0.1
Context window	1024
Training	
Batch size	512
Gradient accumulation steps	32
Effective batch size	16,384
Epochs	3
Learning rate	3×10^{-4}
Weight decay	10^{-5}
Optimizer betas	[0.9, 0.95]
Max gradient norm	1.0
Scheduler	Cosine annealing
Warmup steps	2000
Early stopping patience	100
Data	
Validation ratio	0.001
Test ratio	0.001
Masking probability (p_{mask})	0.5
Expression Encoding	
Raw hidden dimension	512
Hard binning bins	50
Soft binning bins	20
Soft binning hidden dimension	512
Log binning max bins	10
Other	
Random seed	777
Number of data workers	8

Table 2: Training datasets used for model pretraining. Dataset IDs are UUIDs from the curated dataset collection.

Dataset ID	Cells	Dataset ID	Cells	Dataset ID	Cells
7d98cc44-b090-4dc8-804f-2750c84fe9d7	2,489	2f05ab20-a092-4bab-9276-3e0eb24e3fee	38,217	3966ba97-beb8-4d0b-9954-d3775cd2cd61	158,978
2d66790a-6621-4a49-8f0d-4002db5cc98d	4,992	76150f40-1989-4977-9e23-696e72d59d9e	118,672	28ab6eb8-dfa4-4536-9f26-7e06c1b98e8e	25,741
50c4a6d6-940b-4c6a-a376-aea2ae2d3168	21,003	c2a461b1-0c15-4047-9fcb-1f966fe55100	97,499	c838aec3-03ef-4398-b882-0e3912abfff0	1,265,624
cda2c8cd-be1c-42e5-b2cd-162caa1c4ce7	255,901	e40c6272-af77-4a10-9385-62a398884f27	65,088	3a29c3df-b45a-403d-bd76-259640245432	4,992
19e46756-9100-4e01-8b0e-23b557558a4c	66,985	ed11cc3e-2947-407c-883c-c53b043917c3	8,573	61d327d1-2227-4c5f-9367-e3559dc79b07	796
70e4f35b-c98c-45a1-9aa9-2053b07315dd	40,815	715327a6-7978-4896-ba91-69d6b04dbbfb	40,191	eeacb0c1-2217-4cf6-b8ce-1f0fedf1b569	9,337
c5ac3ec2-24b0-43cc-9aab-bb0ebbe205ce	4,992	a6046b15-a095-43b0-9fb5-b36899d87fdb	4,992	0fe5eed4-bcbc-4c00-a388-00bc1455a9b7	4,992
bac09168-940b-4b11-8d55-b4955f80b98d	12,461	f9cfac8d-bff6-47a2-a1f6-503827d375f5	637	43aa19d2-c723-4822-979d-d2f0239835e0	37,121
095940cb-7422-4510-96e2-cbafd961eb88	52,045	30f5e171-83d7-4fc0-bf75-384f122346b3	1,790	3079e9b0-c6db-45c8-b998-a4555a73968e	28,943
79884ae5-e026-4d4a-858f-e807960bd4f7	4,425	489318a0-24c3-4f5c-b105-f084ed0ea026	13,900	0920bcb8-4b3a-4e9d-a353-56f529fd3b32	48,478
d551b400-b2e5-454d-b5a9-ece03e6b4739	4,992	a7b4f565-691d-43ea-bf4a-d2d1d52bb4b4	27,111	93091496-be48-4122-b945-9af9c22a7535	28,718
b9b592d4-a0cc-4694-8704-a6625829ef1f	4,992	34229bdc-a895-4394-8820-574e4028d8c6	31,924	01209dce-3575-4bed-b1df-129f57fbc031	51,876
31f657dc-1875-4c4b-a5ca-ce63b3ef3a82	121,916	e5f5d954-cf0e-4bd8-9346-8d1ddf15a08b	2,487	879bb6df-cc2a-40f1-854b-5be9629d03b2	4,992
edbfa04a-f1dd-496a-8237-df11d70621ca	77,525	06ef6b36-6c9b-4e10-8a94-d0baf274276e	10,533	79527108-1f6c-4152-afe0-1fcdf2e02ba3	4,992
6f0858c0-c590-4740-b022-c152e7608d66	4,992	9ea768a2-87ab-46b6-a73d-c4e915f25af3	40,268	dee75ca4-8348-471e-bbeb-e2143209e3d0	4,992
949ec7fc-ac54-47c9-bb6f-ee9e67688cce	38,937	0bae7ebf-eb54-46a6-be9a-3461cecefa4c	27,675	1df5fa02-4a6e-4b00-a203-cb0a60e75637	4,992
729f397a-0812-4b52-a7d1-b377107ffb41	4,992	53d62b10-bae5-48ac-b16e-71be9ba6de59	4,992	364bd0c7-f7fd-48ed-99c1-ae26872b1042	931,012
ae4552dc-e2ea-4d67-b375-03ec7480f780	37,275	fe2479fd-daff-41a8-97dc-a50457ab1871	292,010	50eb1e23-b8d4-4f76-a184-44e5541fa05a	4,775
77c1c785-809f-4065-8c54-6a0170783256	37,767	49108ba9-1b7a-4a8a-9859-3d32e6a83926	4,992	39ed7d98-676d-4b8d-9d0a-0f3b60914ead	118,647
dd03ce70-3243-4c96-9561-330cc461e4d7	23,732	00ba8341-48ec-4e4e-bb56-be0dd2dd7913	4,992	9ddea8d9-cc4c-420a-90f6-880996f808d4	100,307
9dbab10c-118d-496b-966a-67f1763a6b7d	1,462,702	ca421096-6240-4cee-8c12-d20899b3e005	81,736	731e6380-879f-4b0b-9a1f-2150208852ef	2,065
43245158-5ae1-4e71-a9a6-67eef49c26bc	113,304	f801b7a9-80a6-4d09-9161-71474deb58ae	6,044	54801477-ac3e-47e3-8170-96c5b40d5c10	4,992
43848156-ba94-47b5-8409-7535cea75678	4,992	c5cfa2b7-abb1-4a50-908f-707b54ca606b	14,094	1cf24082-59de-4029-ac81-6e398768af3a	29,522
978a566a-dc27-478d-b306-26daba116c1f	1,102,250	726afd49-df7b-4b56-967a-0fb79d85ee4b	4,992	9adb1b29-65a2-4dd0-86bf-c02690d65fbd	4,992
e6361237-ac4e-4c5d-ad8f-f16aca0c0a8f	66,719	344f27ab-428c-4a0e-a7e1-d4441f2f9b80	4,992	af8b241a-c72c-4470-b1a4-80e7336c6ab6	4,335
346c5aad-b034-4248-8cbe-0a05fd634b9c	163,779	529bb209-9d7b-44da-bfa4-f6e4745c46c2	32,678	33da10b0-9c1d-4c82-9b14-c67cdcf9fae5	30,022
d98ea49e-b70d-4434-b850-bbe217c9b66e	15,216	5e57cd50-8e42-42d6-940d-5c1660d06864	693,682	3e55180c-780a-4424-9434-5296640ffc0d	7,774
f3c49918-4707-4d92-bb6d-2b5b4eb9d1b4	15,177	c5cddbbb-8ba4-4338-8b34-15edb5231e22	4,992	b617ee1b-f8c8-4de9-b82b-e803ab93550d	391,963
9f499d32-400d-4c42-ac9a-fb1481844fee	56,367	93131426-0124-4ab4-a013-9dfbcd99d467	24,327	abd889c6-f60a-4fbd-924e-ee1e9dcf909b	4,992
2f6a20f1-173d-4b8d-860b-c47ffea120fa	2,868	75548d10-160d-4f3e-b317-99ad9630c62d	4,992	e84f2780-51e8-4cfa-8aa0-13bbfef677c7	167,598
be401db3-d732-408a-b0c4-71af0458b8ab	135,462	6e9e3264-02e1-455a-840b-4fbcee132ae7	4,992	639ffc23-14db-4060-9027-3c90314200f8	35,290
4724c395-0c46-46d2-81f7-60fd271fb488	35,350	4fd2ee79-ab3a-4827-a773-1b7dcd099307	4,992	c7775e88-49bf-4ba2-a03b-93f00447c958	647,366
15c5c186-df92-4b17-a253-199e10ffe98a	4,992	019c7af2-c827-4454-9970-44d5e39ce068	12,590	8c42cfd0-0b0a-46d5-910c-fc833d83c45e	65,662
1b767f95-d0a0-4a3d-b394-cc665d86c3dc	34,933	b3c55d0d-4529-4b61-b485-2902e6be0e4e	4,992		
Total: 10,010,835					