Cell2Sentence: Teaching Large Language Models the Language of Biology

Daniel Levine[*]
Yale University
daniel.levine@yale.edu

Syed Asad Rizvi[*]
Yale University
syed.rizvi@yale.edu

Sacha Lévy[*]
Yale University
sacha.levy@yale.edu

Nazreen Pallikkavaliyaveetil
Yale University
nazreen.pm@yale.edu

Ruiming Wu
University of Pennsylvania
wuru@seas.upenn.edu

Insu Han
Yale University
insu.han@yale.edu

Zihe Zheng
Yale University
zihe.zheng@yale.edu

Antonio Henrique de Oliveira Fonseca
Yale University
antonio.fonseca@yale.edu

Xingyu Chen
ETH Zürich
xingyuchen@student.ethz.ch

Sina Ghadermarzi
Yale University
sina.ghadermarzi@yale.edu

Amin Karbasi
Yale University
amin.karbasi@yale.edu

Rahul M. Dhodapkar[†]
University of Southern California
rahul.dhodapkar@med.usc.edu

David van Dijk[†]
Yale University
david.vandijk@yale.edu

Abstract

Large language models like GPT have shown impressive performance on natural language tasks. Here, we present a novel method to directly adapt these pretrained models to a biological context, specifically single-cell transcriptomics, by representing gene expression data as text. Our Cell2Sentence approach converts each cell's gene expression profile into a sequence of gene names ordered by expression level. We show that these gene sequences, which we term "cell sentences", can be used to fine-tune causal language models like GPT-2. Critically, we find that natural language pretraining boosts model performance on cell sentence tasks. When fine-tuned on cell sentences, GPT-2 generates biologically valid cells when prompted with a cell type. Conversely, it can also accurately predict cell type labels when prompted with cell sentences. This demonstrates that language models fine-tuned using Cell2Sentence can gain a biological understanding of single-cell data, while retaining their ability to generate text. Our approach provides a simple, adaptable framework to combine natural language and transcriptomics using existing models and libraries. Our code is available at: https://github.com/vandijklab/cell2sentence-ft.

[*]Co-first authors.

[†]Co-corresponding authors. Correspondence should be addressed to: david.vandijk@yale.edu and rahul.dhodapkar@med.usc.edu.

1 Introduction

Large language models (LLMs) such as GPT have demonstrated powerful capabilities in natural language processing tasks including question answering, summarization, and text generation ([1], [2], [3], [4], [5], [6]). However, applying LLMs to other domains like biology remains an open challenge. In particular, a method to directly apply existing LLMs to single-cell transcriptomics could enable new ways of analyzing, interpreting, and generating single-cell RNA sequencing data. Current methods in this domain rely on specialized neural networks that do not leverage the pretrained knowledge and language understanding of large language models.

In this work, we aim to extend the capabilities of LLMs to the domain of transcriptomics. Our key idea is representing single-cell data in a text format amenable to language models through a method called Cell2Sentence (C2S) [7]. C2S transforms each cell's gene expression profile into a plaintext sequence of gene names (also known as gene symbols) ordered by expression level (Figures 1 and 2). Importantly, we show this rank transformation can be reverted with minimal loss of information (Figures 3 and 7). This allows any pretrained causal language model to be further fine-tuned on cell sequences. Critically, we find that natural language pretraining followed by C2S training significantly improves model performance on transcriptomic tasks compared to training with C2S only, with performance additionally scaling with model size. Our fine-tuned models can generate cells by completing sequences of gene names, generate cells from natural language text prompts, and generate natural language text about cells. By leveraging LLMs' vast pretrained knowledge and combining both modalities, we can enable models that not only generate and interpret transcriptomics data, but also interact in natural language.

Potential applications include inferring how gene expression would change under perturbations, generating rare cell types, identifying gene markers, and interpreting transcriptomics via natural language. Such capabilities could aid biologists and advance single-cell research.

To accomplish this, we first convert single-cell data into C2S format. We then fine-tune pretrained LLMs on these cell sequences using a simple and scalable method. We demonstrate that fine-tuned models can generate accurate cell types and understand transcriptomics data well enough to predict cell labels when prompted. Additional metadata can be provided for conditioning.

We aim to keep our approach simple and as close to standard training pipelines in language modeling as possible. The benefits of adhering to this philosophy are twofold:

1. *Easy to use.* Any user can readily leverage any available pretrained language model by using popular third-party libraries like Hugging Face Transformers [8] that have streamlined model deployment and fine-tuning. The only additional steps needed are to preprocess raw single-cell data into C2S format, and to postprocess generated texts back into gene expression vectors, both done seamlessly using our modular data transformation pipeline provided in our GitHub repository.
2. *Easy to modify.* The simplicity of the training pipeline means the user can mold the preprocessing and training steps to their liking, e.g. by adding more metadata or prepending specific prompts.

In summary, our key contributions are:

1. Introducing Cell2Sentence, an effective method for representing single-cell data as text sequences.
2. Showing large language models can be fine-tuned on cell sentences to generate cells and classify cell types.
3. Providing a simple and modular framework for adapting LLMs to transcriptomics using popular LLM libraries.

In the following manuscript, we first explain the C2S data transformation and model fine-tuning process. We then evaluate fine-tuned LLMs on tasks including cell generation, classification, and expression recovery to demonstrate their biological understanding. Finally, we discuss the implications and future directions for combining natural language and transcriptomics through representation learning.

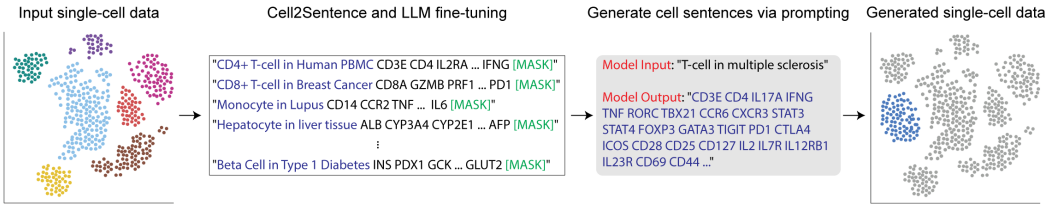

Figure 1: Overview of the Cell2Sentence framework. Input single-cell data, including metadata, are converted into cell sentences for LLM fine-tuning. Inference, via prompting, generates new cell sentences that can be converted back to gene expression space.

1.1 Related Work

Large language models Large language models have been adapted for a wide range of tasks on textual data in the field of natural language processing. Some key tasks and sample architectures include text classification (LSTM [9], BERT [3], RoBERTa [10]), question answering (LlaMA-2 [5], Falcon [11]), and text generation (T5 [12], GPT-3 [13], BART [14]). For a more in-depth discussion of developments around large language models, the reader is directed to [15].

Single-cell foundation models Alongside the development of large models for natural language processing, deep neural networks have been developed to accomplish numerous tasks on single-cell transcriptomics data. Architectures have been described for several tasks inluding: cellular annotation—where a cell is assigned a label according to it's biological identity (NeuCA [16], ACTINN [16], scVI [17])—, batch effect removal/sample integration—where transcript abundance differences due to technical replicates are removed (scVI, scGen [18], SAUCIE [19])—, and data imputation—where missing transcript abundance data are inferred (scVI, DeepImpute [20], SAUCIE).

More recently, efforts such as the Gene Expression Omnibus (GEO) [21] and Human Cell Atlas (HCA) [22] have centralized and standardized the storage of data from hundreds of single-cell experiments across a wide range of tissues, comprising hundreds of millions of measurements. Several models have been designed and trained on this data (e.g. scGPT [23], scFoundation [24], Geneformer [25]), with the goal of creating a foundation model for single cell transcriptomics data analogous to foundation models in natural language processing.

Prompt fine-tuning Since the introduction of GPT-2 [26], prompting has become a common method to elicit meaningful behavior from large language models ([27], [28], [29]). Recently, publicly available datasets like Alpaca [30] and dataset generators such as Flan [31], [32], [33] together with parameter efficient fine-tuning [34] have made it possible to train custom large language models. A survey of methods can be found in [35].

Multimodal training and cross-modality encoding Unlike most multimodal machine learning schemes, which perform encoding of each modality separately either to a shared latent space, or parallel latent spaces coordinated by a constraint, our C2S framework transforms data directly into a single format (text) *prior* to embedding [36, 37]. One previously described analogous approach is the visual language modeling strategy of [38], which transformed images into a visual bag-of-words format prior to classification. However, to our knowledge, no such analogous approaches have been described using modern embedding strategies and in the domain of single-cell transcriptomics.

2 Results

In this section, we present several benchmarks and evaluations to demonstrate the different use cases of Cell2Sentence. All presented models are trained on a human immune tissue dataset [39] (see Section 3). "NL + C2S" models are pretrained on natural language and then fine-tuned on cell sentences. "C2S" models are only trained on cell sentences.

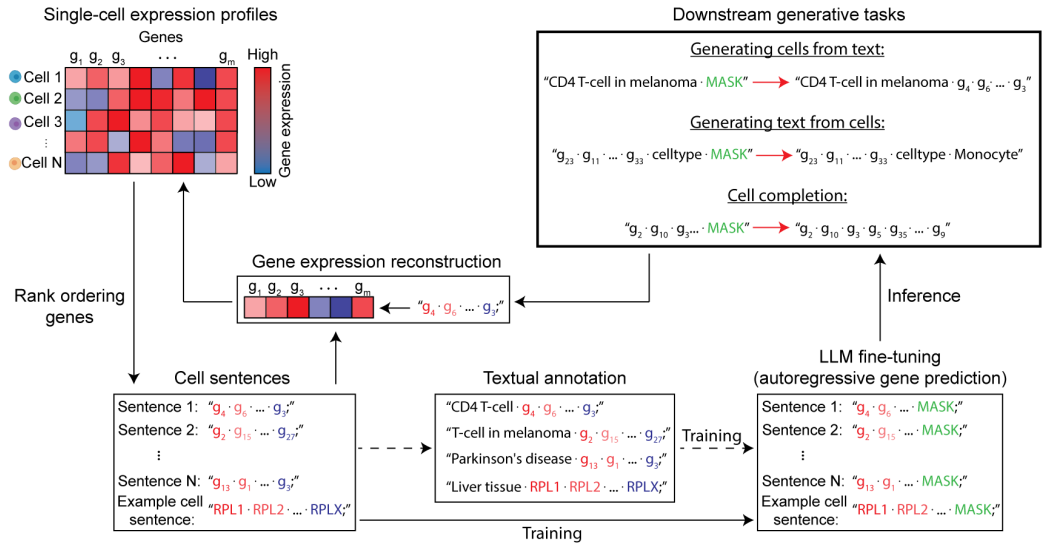

Figure 2: Detailed overview of the Cell2Sentence framework. Single-cell gene expression profiles are transformed into cell sentences via expression rank orderering of gene names. Cell sentences may be annotated with biological metadata, such as cell type, tissue, or disease. LLMs are then fine-tuned on the cell sentences. Inference is done by generating cells via autoregressive cell completion, generating cells from text, or generating text from cells. The resulting generated cell sentences can be converted back to gene expression.

2.1 Cell sentence encoding is a robust and reversible operation

Gene rank and gene expression follow a log-linear relationship in scRNAseq data. Single-cell RNA sequencing produces transcript count matrices that represent the genetic profiles of individual cells. Most current computational models in single-cell biology concentrate on handling data in $\mathbb{R}^{c \times n}$, posing scalability challenges with larger datasets. We propose transforming expression matrices into gene sequences as a solution to enable the use of LLMs [40, 41] and other transformer-based architectures [42] for single-cell data analysis.

While genes are not intrinsically ordered in transcript matrices, their expression patterns have been shown to follow inverse-rank frequency patterns [43, 44], thus establishing a steady relationship between a gene's expression level within a cell and its rank among the genes expressed in that cell. We model this inverse-rank relationship with a log-linear distribution and approximate it in log-log space using a linear regression [7].

The resulting models allow us to convert cells back and forth between gene rank and expression domains (see Section 3.4). We leverage this capability to produce rank-ordered sequences of gene names which we use to train our subsequent language models (see Section 3.1). We present visualizations for the relationship between normalized gene expression and rank on a variety of human tissue datasets in Appendix Figure 7, as well as additional metrics on the performance of gene expression reconstruction in Appendix Figure 8.

C2S enables forward and reverse transformation with minimal information loss. We first evaluate the effectiveness of our C2S transformation to reconstruct the original gene expression of a cell from cell sentence format. Figure 3A visualizes the reconstruction performance of a fitted linear regression on an immune tissue dataset comprising 50K cells and over 35K genes. A simple linear model captures over 81% of the variation in the gene expression, requiring only the log rank value of that gene for expression reconstruction. This demonstrates that the transformation to cell sentences and back to expression space preserves much of the important information in single-cell data. This allows for analysis in the natural language space of cell sentences followed by accurate conversion back to expression. Figures 3B and 3C visualize the original ground truth immune tissue data alongside reconstructed expression data from converted cell sentences. This qualitatively shows that important structure in the immune tissue data as well as cell type separation are retained.

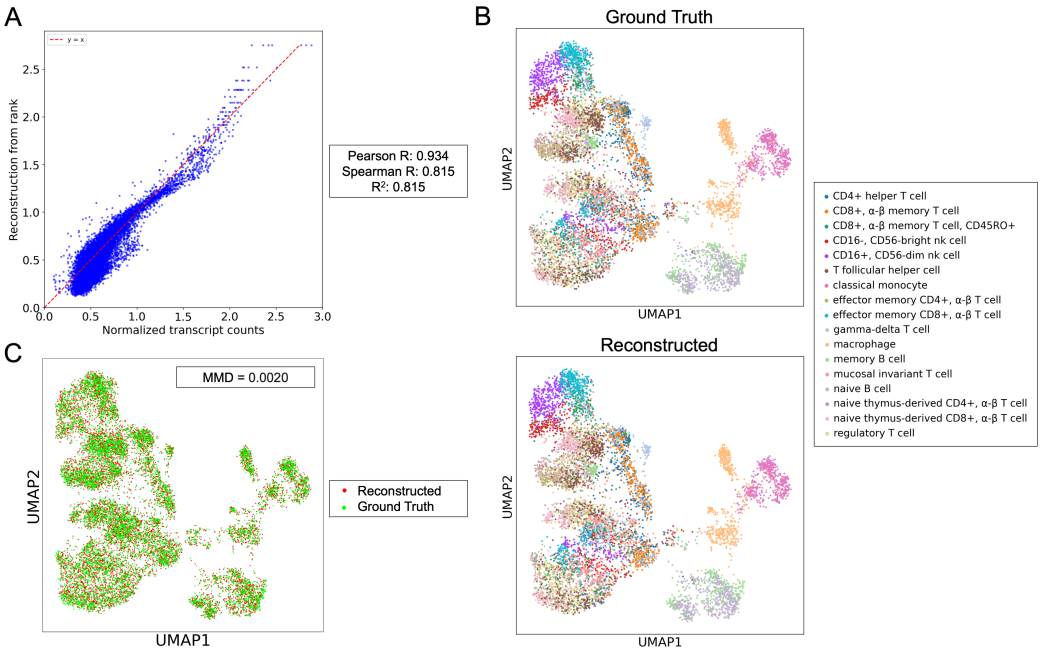

Figure 3: Cell2Sentence transformation can accurately reconstruct cell expression from cell sentences. (A) Expression reconstruction from rank using a linear model. (B) UMAP of ground truth expression (top) versus reconstructed expression from cell sentences (bottom). (C) UMAP of ground truth expression (green) and reconstructed expression from cell sentences overlaid.

Model	Pearson R	Spearman R	R^2
GPT-2 Small (C2S)	0.947	0.766	0.873
GPT-2 Small (NL + C2S)	**0.984**	0.768	**0.949**
GPT-2 Medium (C2S)	0.948	**0.781**	0.88
GPT-2 Medium (NL + C2S)	0.982	**0.781**	0.943

Table 1: Correlation metrics for averaged generated cells per cell type against original expression values. "NL + C2S" means pretrained on natural language and then trained on cell sentences. "C2S" means no pretraining and just trained on cell sentences. Correlation metrics are computed per cell type, and are averaged across all 17 cell types, forming a score for the quality of an average generated cell.

2.2 LLMs can meaningfully manipulate cells as text

LLMs trained on cell sentences show healthy convergence behavior. We find that infusing sequential structure to single-cell data yields a non-trivial and meaningful textual modality. We train four GPT-2 models [26] on a corpus of cell sentences sampled from [39]. While we pretrain models exclusively on cell sentences, we also experiment with fine-tuning pretrained GPT-2 models. We find that all our models learn cell sentence distributions and converge during training with a steadily decreasing perplexity (see Figure 5). In both settings, we find that larger models achieve lower perplexities. Overall, we demonstrate the capability of causal language models to learn the cell sentence semantic distribution from a narrow scRNA-seq dataset.

Cell reconstruction: R^2 gene prediction. Accurate generation of different cell types is crucial for generative approaches on single-cell data, as it enables downstream analysis using the model. To evaluate the ability of our trained models to generate realistic cells, we consider the average generated cell of each of the 17 cell types in our immune tissue dataset, and compare it with the average real cell of each cell type in Table 1. Across 17 different cell types, generated cells from finetuned models show high correlation with real cells, capturing over 94% of the variation in the expression of an

average cell. We note that initializing a model with a pretrained language model outperforms training from scratch, indicating that there is mutual information which allows a model to better understand cell sentence generation.

Generate cells from text: generate cells from cell type label. We assess the model's performance by calculating the k-Nearest Neighbors (KNN) accuracy for generated cells using two distinct methods:

1. Classify the generated cell type based on the types of its nearest neighbors in the ground-truth dataset.
2. Classify the generated cell type based on the types of its generated neighbors.

The label assigned to a generated cell corresponds to the cell type used for its conditional generation, as detailed in Section 3.2. Predictions of type 1 determine if the model is capable of approximating real cells within the corresponding cell type, whereas predictions of type 2 show the model is capable of generating distinct cell types. The UMAP plots in Figure 4 show how much separation can be achieved by representing cells with the 100 highest expressed genes. Unlike reducing the size of the gene expression matrix by selecting a subset of highly variable genes, our approach allows different cells to be encoded with the specific genes that are most relevant to them, potentially allowing for better representation of rare cell types at similar levels of compression. Additionally, since the immune dataset has many sub-types for some cell types (e.g. T-cells), we expect some sub-types to be very close when restricted to the 100 genes with the highest expression. Still, nontrivial structure emerges in all of the plots.

In Figure 4C and D, the UMAP plots demonstrate that the reconstructed gene expression from the top 100 generated genes not only maintains the general structure of the original data but also closely aligns with cell type-specific distinctions in the baseline UMAP visualizations. This validates our generative model's ability to capture both macroscopic and fine-grained expression profiles, attesting to its efficacy in creating biologically relevant cellular representations.

To further corroborate the gene expression patterns of our generated cells, we employ gene expression heatmaps, presented in Figure 10. These heatmaps compare the differentially expressed genes for each cell type in both the generated and ground-truth cells, with the differential genes sourced from the top 100 genes in the ground truth dataset. The heatmaps confirm that the generated data is consistent with the overall structure in the original dataset while also highlighting subtle gene-level discrepancies. This enhances our understanding of how well the generated cells mimic the intricacies of the baseline data. Additionally, the analysis points to promising avenues for using generated cells to identify key marker genes, offering critical insights into cellular function and laying the groundwork for future research.

Model	K=5		K=10		K=25		K=50	
	Expr.	Lev.	Expr.	Lev.	Expr.	Lev.	Expr.	Lev.
Real cells (ground truth):	62.60	37.55	62.32	39.86	60.45	42.64	58.31	44.33
Generated cells:								
scVI	43.06	-	44.24	-	45.06	-	43.63	-
GPT-2 Small (C2S)	24.56	16.02	23.79	17.72	23.51	19.17	22.84	19.76
GPT-2 Small (NL + C2S)	52.55	37.63	52.19	41.20	51.42	43.42	49.51	44.44
GPT-2 Medium (C2S)	26.36	17.58	25.48	19.18	24.55	20.67	23.44	21.32
GPT-2 Medium (NL + C2S)	**54.67**	**38.60**	**54.52**	**41.34**	**53.27**	**43.93**	**51.80**	**44.73**

Table 2: KNN classification accuracy results against ground truth data. "NL + C2S" means pretrained on natural language and then trained on cell sentences. "C2S" means no pretraining and just trained on cell sentences. KNN classifier is fitted on ground truth cell sentences and used to predict the cell type label of generated cell sentences from different trained models. KNN classification is done both in cell sentence space using Levenshtein distance (Lev.), as well as after converting back to expression vectors (Expr.). "Real cells" indicates KNN classification fit on ground truth cell sentences and used to predict a separate sample of ground truth cell sentences.

Tables 2 and 3 shows the KNN accuracy of each model for different number of neighbors. Generated cells from our trained models are accurately classified by cell type by a KNN classifier 2, with up

Model	K=5		K=10		K=25		K=50	
	Expr.	Lev.	Expr.	Lev.	Expr.	Lev.	Expr.	Lev.
scVI	45.72	-	47.04	-	45.36	-	39.90	-
GPT-2 Small (C2S)	28.65	70.61	30.03	52.26	31.68	40.88	32.66	36.21
GPT-2 Small (NL + C2S)	60.07	76.09	60.41	66.95	60.55	60.04	59.42	56.80
GPT-2 Medium (C2S)	32.28	69.57	34.00	54.50	34.57	43.93	33.78	39.71
GPT-2 Medium (NL + C2S)	**62.95**	**76.67**	**62.98**	**67.77**	**63.09**	**60.95**	**62.05**	**57.51**

Table 3: KNN classification results for separation of distinct cell types. "NL + C2S" means pretrained on natural language and then trained on cell sentences. "C2S" means no pretraining and just trained on cell sentences. This measures the model's ability to generate distinct clusters when conditioned on cell type. This is done both in cell sentence space using Levenshtein distance (Lev.), as well as after converting back into expression vectors (Expr.).

to 54% classification accuracy. The distinctness of generated cells is further quantified in Table 3, showing that our models are capable of generating distinct cell types through natural language. We compare against scVI [45], a probabilistic generative model for single-cell data based on variational inference. We observe that our models outperform scVI in terms of generation quality against the ground truth cells. Additional UMAPs of generated cells from scVI are available in Figures 13 and 14. We require cell types to have at least 500 samples in the training dataset so the number of training samples per cell type is large enough to be meaningful for comparison. This filtering condition should be dropped for larger training datasets. See Sections 2.2 and 3.4 for details on transforming cell sentences to expression values.

Generate text from cells: autoregressive cell type prediction. Models pretrained with natural language (NL) can be trained with Cell2Sentence to generate meaningful text from cells. Our results illustrate the efficacy of this approach through autoregressive prediction of cell types. Note this is distinct from traditional classification—we do not train a classifier head or modify the architecture in any way. We simply prompt the model with a cell sentence and ask it to identify its cell type in natural language (see Figure 6).

Table 4 underscores the necessity of NL pretraining for accurate cell type identification. Figure 12 presents the corresponding confusion matrices. A significant performance decline is observed when using models that have not undergone NL pretraining, thereby confirming that the models are not merely memorizing the conditioning text. Despite 1) the limited scope of natural language text in our training prompts relative to the pretraining corpus, and 2) permitting the models to train on these natural language prompts, models without NL pretraining failed to acquire meaningful natural language embeddings. Furthermore, a modest performance increment is observed as the scale of pretrained models increases.

Model	Accuracy	F1	Precision	Recall
Random Classifier	2.86	2.69	2.86	2.57
GPT-2 Small (C2S)	29.33	17.46	13.27	29.33
GPT-2 Small (NL + C2S)	69.95	68.56	69.44	69.95
GPT-2 Medium (C2S)	29.17	17.41	13.35	19.17
GPT-2 Medium (NL + C2S)	**74.26**	**73.69**	**74.16**	**74.26**

Table 4: Quantification of autoregressive cell type prediction on unseen cells. "NL + C2S" means pretrained on natural language and then trained on cell sentences. "C2S" means no pretraining and just trained on cell sentences. These results show test accuracy significantly improves with NL pretraining. The scores are computed on unseen immune tissue test data and weighted by the distribution of labels.

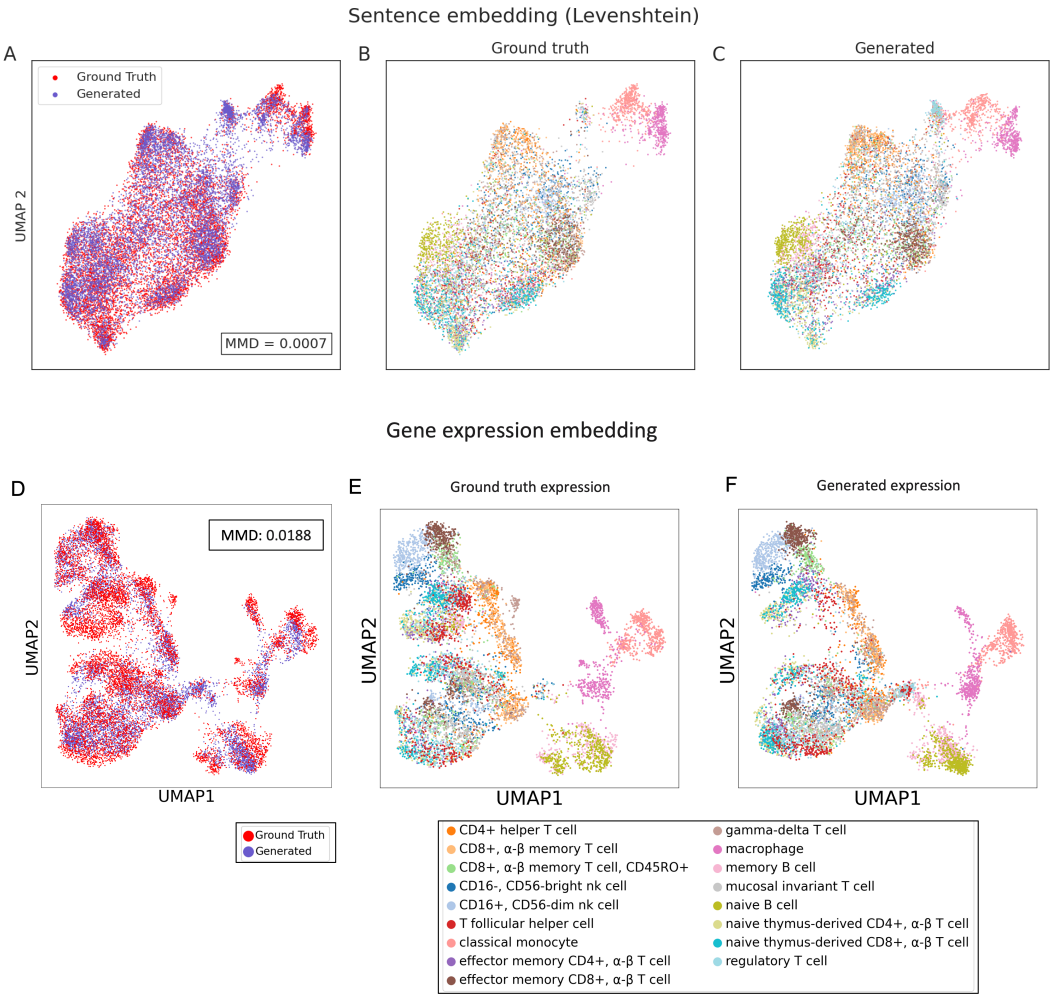

Figure 4: Plots for cell sentences and expression data. The UMAP plots show that models pretrained with language and fine-tuned with Cell2Sentence achieve quality generated outputs in the sentence space. We see good cell type separation for both generated and ground truth cell sentences in addition to good overlap, showing that our generated sentences model the ground truth distribution well. Plots A, B, and C are UMAP plots of generated cell sentences versus real cell sentences using Levenshtein distance. We computed the Maximum Mean Discrepancy (MMD) [46] statistic with the Python library scMMD [47]. Plots D, E, and F are UMAP plots of generated cell expression vectors and ground truth cell expression vectors. See Table 2 and Table 3 for a quantification of these plots.

3 Methods

3.1 Data transformation

The Cell2Sentence transformation is at its core a reorganization of the cell expression matrix into sequences of gene names ordered by decreasing transcript abundance, similar to rank-ordering transformations of count matrices [44]. Let C denote a cell by gene count matrix with n rows and k genes, with $C_{i,j}$ denoting the number of RNA molecules observed for gene j in cell i. We follow standard preprocessing steps for single-cell RNA seq data, including filtering cells with fewer than 200 genes expressed and filtering genes which are expressed in fewer than 200 cells. Quality control metrics are then calculated based on mitochondrial gene counts within each cell using the Scanpy Python library [48], and low-quality cells are filtered out which contain over 2500 counts, or which have greater than 20 percent of transcript counts from mitochondrial genes. The count matrix is then

row-normalized so that each cell sums up to 10,000 transcript counts and then log-normalized [49], obtaining the final preprocessed count matrix C'. We summarize this normalization step as:

$$C'_{i,j} = \log_{10}\left(1 + 10^4 \times \frac{C_{i,j}}{\sum_{j=1}^{k} C_{i,k}}\right) \tag{1}$$

We denote the rank-order transformation applied on C' as S, and the sequence of gene names resulting from $S(C_i)$ as cell sentence s_i for each cell i in the preprocessed count matrix. In practice, we apply the preprocessing and rank-order transformation S on each individual single-cell dataset, providing a flexible process for converting traditional single-cell gene expression count matrices to cell sentences.

We construct our dataset by sampling 49,920 cells from a large dataset of human immune tissue cells [39]. We apply the previously outlined normalization steps and convert to cell sentences. We split the resulting cell sentences into training (39,936), validation (4,992), and test (4,992) sets. We also attach each cell's type as specified in [39]. In order to limit computational costs, we truncate each cell sentence to only keep the 100 highest expressed (top 100 ranked) genes. Note that this truncation operation minimizes the resulting ordering variability, as genes with lower rank have more similar expression values. We consider this as a future point of study.

3.2 Training

We trained a total of four GPT-2 models for this study. Specifically, we pretrain and fine-tune both GPT-2 small and medium on our cell sentence corpus. In both settings, we format inputs to the model as prompts, providing natural language context for the model to learn from (Figures 1 and 2). We follow a standard training configuration and use the AdamW optimizer [50]. We leverage half-precision floating points (FP16) and gradient accumulation for memory savings. We found negligible improvement with using full-precision floating points (FP32) at the cost of a 60% slowdown.

3.2.1 Tasks

As outlined in Section 2, we formulate three downstream tasks (see Figure 6):

1. Unconditional cell sentence generation: produce a sequence of 100 genes without any prescribed cell type label.
2. Conditional cell sentence generation: generate a sequence of 100 genes given a specific cell type label.
3. Cell type prediction: generate a cell type label following a provided sequence of 100 genes.

During the training phase, for each iteration, we randomly select a task and subsequently pick a corresponding prompt template from a set of approximately 20 templates per task. These templates, while varied in phrasing, retain consistent semantic meaning. The prompt structure for cell type prediction combines the *prompt* with the *cell sentence*. For conditional cell generation, it merges the *prompt* with the specified *cell type*. In contrast, the unconditional cell generation prompt primarily consists of a succinct directive, as depicted in Figure 6.

3.2.2 Pretraining

The GPT-2 small model is initialized with 12 layers and 768 hidden dimensions, and the medium model with 24 layers and 1024 hidden dimensions, as detailed in [26]. We employ a learning rate of 6×10^{-4} with a cosine scheduler and 1% warmup ratio. For the GPT-2 medium model, we accumulate gradients over 16 steps. The effective batch sizes for the small and medium models are of 10 and 48 examples. Each model is trained using a single A5000 GPU over two days.

We train a Byte Pair Encoding (BPE) tokenizer [51] on the full cell sentence dataset, including NL prompts and cell type labels, yielding a vocabulary of 9,609 tokens. The training set contains approximately 30 million tokens, averaging 740 tokens per example. Due to the smaller embedding space, the initialized models contain slightly fewer parameters than their counterparts pretrained on a vocabulary of 50,257 tokens (93M for the small model and 313M for medium model, as shown in Figure 5). The resulting corpus exhibits sparse NL tokens due to short and repetitive prompts. Despite instruction corpora being traditionally used to fine-tune pretrained models for question

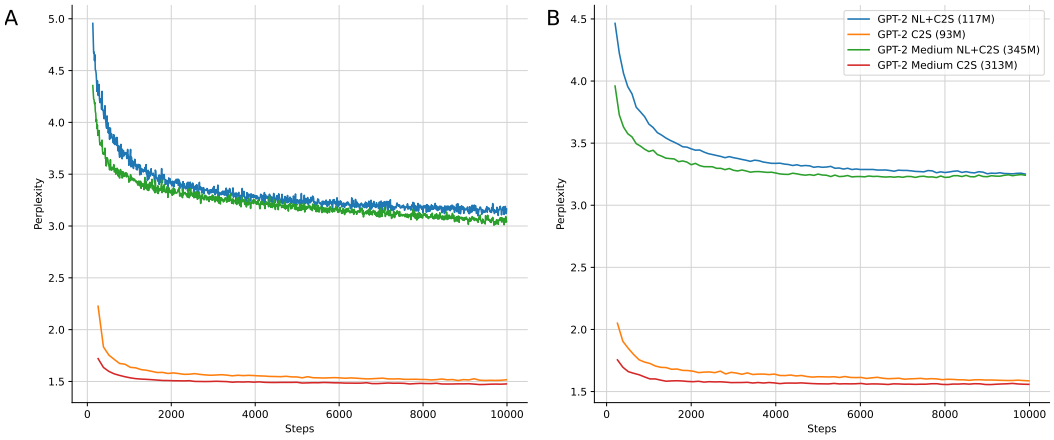

Figure 5: Perplexity curves for finetuned and pretrained models. (A) Estimated model perplexity computed after the training loss. (B) Estimated model perplexity computed on the validation set during training. All models preserve the default GPT-2 context length of 1024 tokens.

answering tasks, we adopt this setting during pretraining to mirror our fine-tuning setup described in Section 3.2.3. We hypothesize that the semantic variability from prompting patterns might implicitly regularize token and positional embeddings, with natural language tokens acting as class tokens.

We emphasize that the loss is computed on both the prompt and the associated label (i.e. cell type). Not doing so would cause embeddings of the prompt tokens to remain random, impairing the capacity of the model to learn the conditional relations between prompt and label tokens. We evaluate the capacity of our model to generate valid genes and maintain an accurate sequence length (here, of 100 genes) and present the results in Table 5. We find that both pretrained models are able to generate sequences of 100 genes without significantly deviating from the mean. The models also both achieve over 97% and 96% accuracy in gene validity and uniqueness.

3.2.3 Fine-tuning

Both models are initialized using pretrained weights from the Hugging Face model hub [52]. We employ a learning rate of 5×10^{-5} with a linear scheduler. On both models, we accumulate gradients over 16 steps and use batch sizes of eight examples (yielding an effective gradient update batch size of 128 examples). Each model is trained using a single A5000 GPU. While we experimented with applying efficient fine-tuning techniques (e.g. LoRA [34]), fully fine-tuned models outperformed alternatives in gene uniqueness and validity assessments. We notably found LoRA to yield highly variable generation patterns, with uniqueness of genes in generated sentences as low as 70%. Unlike for our pretraining setup, we apply the instruction fine-tuning task in a classical manner, computing the loss exclusively on labels. We use the pretrained GPT-2 tokenizer, which averages around 233 tokens per training samples (yielding a total of 9M training tokens).

Similarly to Section 3.2.2, we examine the coherence of generated output using sequence length, as well as accuracy in gene validity and uniqueness. We find that the fine-tuned model outperform the pretrained models by generating genes with over 99% validity and 98% uniqueness on average. While both models achieve reliable performance by these standard metrics, we conclude that our fine-tuned models are consistently outperforming the pretrained models in generating real human genes, which are only rarely duplicated within cell sentences.

3.3 Inference

At inference, we follow the training procedure in Section 3.2 (see Figures 1, 2, and 6). We set the hyperparameters top_p $= 0.95$ and top_k $= 50$ to promote diversity in cell generation. For conditional cell generation and unconditional cell generation used in our experiments from Section 2.2, we randomly sample prompt templates as input, inserting the cell type where needed for

Model	Gen. Length	Valid Genes %	Unique Genes %
GPT-2 Small (C2S)	101.54	97.84	97.31
GPT-2 Small (NL + C2S)	99.84	99.60	98.88
GPT-2 Medium (C2S)	100.62	97.51	96.69
GPT-2 Medium (NL + C2S)	99.80	**99.70**	**99.47**

Table 5: Quality of generated outputs. "NL + C2S" means pretrained on natural language and then trained on cell sentences. "C2S" means no pretraining and just trained on cell sentences. This table shows 1. models trained using Cell2Sentence are able to generate real genes with few duplicates and nonsense genes and 2. models pretrained with natural language generate more accurately. The metrics are computed across all 35 cell types seen during training with 500 cells generated per cell type and then averaged across all generated cells (top 100 genes). The valid genes percentage shows the number of genes generated that are real genes including duplicates. The generated length is the number of genes generated regardless of their validity. The unique gene ratio is the ratio of unique valid genes to the generated length.

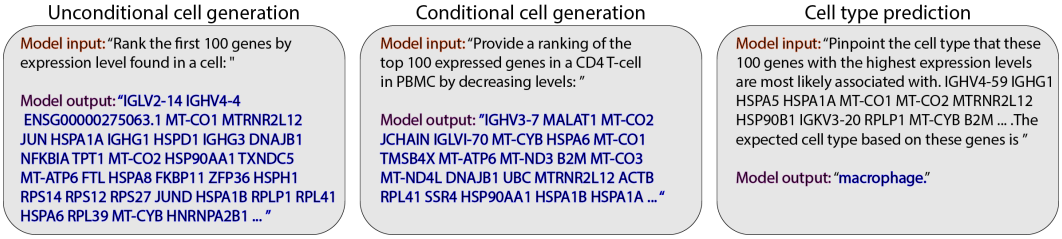

Figure 6: Depiction of three types of prompts used during training and generation. From left to right: unconditional cell generation, conditional cell generation (e.g.: with cell type), and autoregressive cell type prediction. For our setup, we generate and prompt with 100 genes.

conditional generation. For autoregressive cell type prediction, we randomly sample templates as in training but use either real cell sentences from our test dataset or generated cell sentences.

All outputs are generated until an end-of-sequence (EOS) token is predicted, which was appended to all training samples. We have experimented with using non-special token delimiters such as semicolons and periods but found them unnecessary for our study's scope. Post-generation, gene and cell type extraction is done using regex to remove prompts. For evaluation, we retain invalid genes and average ranks of duplicate genes, rearranging sequences as needed. When reverting back to expression values, invalid genes are ignored, but the rank values are preserved, e.g. if an invalid gene appears in position 3 and a valid gene appears in position 4, the invalid gene is ignored, but the valid gene retains a rank of 4.

Utilizing pretrained LLMs offers the advantage of using highly optimized, open-source libraries for inference. We make use of flash attention [53, 54] and batched inference to accelerate generation. Inference for GPT-2 medium (345M parameters) can be done on a single A5000 GPU with 24GB of VRAM and a batch size of 100 without running out of memory. For GPT-2 small (117M parameters), the batch size can be increased to 250. We did not determine the exact maximum batch sizes, so these values can likely be increased further. On average, the number of tokens in the prompts and top 100 genes combined was around 350. For example, the GPT-2 small model takes approximate 20 minutes to generate 500 cells from each of the 35 cell types found in the immune tissue dataset. Model quantization was not required but may be useful for future experiments with larger models.

3.4 Transformation back to expression

To transform generated cell sentences back to expression space, we define a simple inverse transformation function using a linear model to predict the expression of the generated gene based on its rank. For a given single-cell dataset which underwent rank-order transformation S, let r_i denote the log of the rank of gene i in C', and e_i the original expression of gene i. We first fit a linear model to predict e_i from r_i during the initial conversion to cell sentence format, resulting in a fitted slope and

intercept value which are saved for each converted dataset (see Figures 7, 8, and 9). Hence, we fit a linear regression of the form $e_i = a_d \times r_i + b_i$, given dataset d and $\{a_d, b_d\} \in \mathbb{R}^2$.

Generated cell sentences output by different models are first run through a postprocessing procedure which replaces any invalid generated gene names with a special ignore token, and averages the rank of any duplicate genes in the generated cell sentence. The fitted linear model parameters are then applied to the log of the rank of the generated genes to convert the sequence of genes back to an expression vector. Note that any genes which are not present in the generated cell sentence are considered to have zero expression, and are filled with zeros in the resulting expression vector. We define the average rank of a generated gene g_i^{gen} belonging to the set of unique genes $G_U \subseteq S$ as follows:

$$r_i^{\text{gen}} = \frac{1}{|G|} \sum_{j=1}^{|G|} \text{rank}(g_j^{\text{gen}}) \tag{2}$$

where $G = \{g_1^{\text{gen}}, g_2^{\text{gen}}, \ldots, g_n^{\text{gen}}\} \subseteq S$ is the set of duplicate generated genes for g_i^{gen}, and r_i^{gen} denotes the average rank of gene g_i^{gen} in the generated cell sentence. This yields the following formulation for expression value vector for the generated cell

$$e_i^{\text{gen}} = \begin{cases} a_d \times \log(r_i^{\text{gen}}) + b_d & \text{if } g_i^{\text{gen}} \in G \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

In practice, we consider a global dictionary of all gene names seen in single-cell datasets, which dictates the size of the resulting gene expression vector of the cell.

4 Discussion

In this work, we presented Cell2Sentence, a new approach for facilitating the training of large language models on single-cell transcriptomics data. The method represents gene expression profiles in single cells as text sequences, which we term "cell sentences". These cell sentences are composed of gene names sorted by their expression levels, thereby creating a robust and reversible encoding of the biological data.

Our investigations revealed that cell sentences conveniently and correctly encode gene expression data in a format easily digestible by existing language models. Language models fine-tuned on these cell sentences not only converge robustly, but also perform significantly better on tasks related to cell sentences as compared to models trained from scratch or other state-of-the-art deep learning models purpose-built for handling single-cell RNA sequencing data. Cell sentences can be seamlessly integrated with textual annotation to perform both generation and summarization tasks, both of which benefit from natural language pretraining. In fact, there are no theoretical limitations on the application of *any* text-based architecture to Cell2Sentence-generated cell sentences. Our findings highlight the benefits of transfer learning in this interdisciplinary setting.

While our research showcases the potential of leveraging language models for transcriptomics, there are several limitations that need to be acknowledged. Cell2Sentence captures gene expression data through a set of gene names sorted by their expression levels. This approach results in the loss of specific quantitative information about the levels of gene expression, requiring the model to infer the strength of expression from gene order alone. While our experiments suggest that the transformation between gene order and transcript counts is robust, there may be more information loss in certain conditions, such as experiments with a very high depth of sequencing and a large number of transcripts recovered per gene. It is worth noting that the natural language output from models fine-tuned using Cell2Sentence cannot be used directly without specific post-processing (e.g. to ensure that generated gene names are meaningful, and that genes are not duplicated in a sentence), though we have shown that, at least in our evaluations, such invalid output is exceedingly rare.

Cell sentences have different characteristics and grammar than natural language, and model architectures that have been designed to perform well on natural language such as GPT-2 may struggle with certain aspects of the structure of cell sentences. Unlike natural language sentences, which are typically composed of tens of individual words, cell sentences are composed of thousands of gene names. As resource requirements for most transformer-based architectures scale with the number of tokens processed in parallel, cell sentences in their original form may be computationally expensive to generate, and/or may require the use of language models optimized for longer sequence

lengths. We believe that processing of cell sentences will benefit greatly from ongoing work in long-sequence/long-context language processing.

Finally, our experiments, while exciting, focus only on a highly restricted set of cell types and tasks. Extending our method to more complex and heterogeneous datasets, as well as to additional tasks enabled by the direct application of existing natural language model architectures is critical to validate the versatility and reliability of Cell2Sentence. In summary, our study serves as a foundational effort that opens the door for the integration of language models with other computational biology techniques.

References

[1] Ashish Vaswani et al. "Attention is All you Need". In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc., 2017.

[2] Alec Radford et al. "Improving language understanding by generative pre-training". In: (2018).

[3] Jacob Devlin et al. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 4171–4186. DOI: 10.18653/v1/N19-1423.

[4] OpenAI. *GPT-4 Technical Report*. 2023. arXiv: 2303.08774 [cs.CL].

[5] Hugo Touvron et al. *Llama 2: Open Foundation and Fine-Tuned Chat Models*. 2023. arXiv: 2307.09288 [cs.CL].

[6] Rohan Anil et al. *PaLM 2 Technical Report*. 2023. arXiv: 2305.10403 [cs.CL].

[7] Rahul M. Dhodapkar. "Representing cells as sentences enables natural-language processing for single-cell transcriptomics". In: *bioRxiv* (2022). DOI: 10.1101/2022.09.18.508438. eprint: https://www.biorxiv.org/content/early/2022/09/19/2022.09.18.508438.full.pdf. URL: https://www.biorxiv.org/content/early/2022/09/19/2022.09.18.508438.

[8] Thomas Wolf et al. *HuggingFace's Transformers: State-of-the-art Natural Language Processing*. 2020. arXiv: 1910.03771 [cs.CL].

[9] Sepp Hochreiter and Jürgen Schmidhuber. "Long Short-Term Memory". In: *Neural Computation* 9.8 (Nov. 1997), pp. 1735–1780. ISSN: 0899-7667. DOI: 10.1162/neco.1997.9.8.1735.

[10] Yinhan Liu et al. *RoBERTa: A Robustly Optimized BERT Pretraining Approach*. 2020. URL: https://openreview.net/forum?id=SyxS0T4tvS.

[11] Ebtesam Almazrouei et al. "The Falcon Series of Language Models: Towards Open Frontier Models". In: (2023).

[12] Colin Raffel et al. "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer". In: *Journal of Machine Learning Research* 21.140 (2020), pp. 1–67. URL: http://jmlr.org/papers/v21/20-074.html.

[13] Tom Brown et al. "Language Models are Few-Shot Learners". In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle et al. Vol. 33. Curran Associates, Inc., 2020, pp. 1877–1901. URL: https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

[14] Mike Lewis et al. "BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, July 2020, pp. 7871–7880. DOI: 10.18653/v1/2020.acl-main.703. URL: https://aclanthology.org/2020.acl-main.703.

[15] Wayne Xin Zhao et al. *A Survey of Large Language Models*. 2023. arXiv: 2303.18223 [cs.CL].

[16] Z. Li and H. Feng. *NeuCA: NEUral network-based single-Cell Annotation tool*. R package version 1.6.0. 2023.

[17] R. Lopez et al. "Deep generative modeling for single-cell transcriptomics". In: *Nat Methods* 15.12 (2018), pp. 1053–1058. DOI: 10.1038/s41592-018-0229-2.

[18] M. Lotfollahi, F.A. Wolf, and F.J. Theis. "scGen predicts single-cell perturbation responses". In: *Nat Methods* 16.8 (2019), pp. 715–721. DOI: 10.1038/s41592-019-0494-8.

[19] M. Amodio, D. van Dijk, K. Srinivasan, et al. "Exploring single-cell data with deep multitasking neural networks". In: *Nat Methods* 16 (2019), pp. 1139–1145. DOI: 10.1038/s41592-019-0576-7.

[20] C. Arisdakessian, O. Poirion, B. Yunits, et al. "DeepImpute: an accurate, fast, and scalable deep neural network method to impute single-cell RNA-seq data". In: *Genome Biol* 20 (2019), p. 211. DOI: 10.1186/s13059-019-1837-6.

[21] Tanya Barrett et al. "NCBI GEO: archive for functional genomics data sets—update". In: *Nucleic Acids Research* 41.D1 (Nov. 2012), pp. D991–D995. ISSN: 0305-1048. DOI: 10.1093/nar/gks1193. eprint: https://academic.oup.com/nar/article-pdf/41/D1/D991/3678141/gks1193.pdf. URL: https://doi.org/10.1093/nar/gks1193.

[22] Aviv Regev et al. "Science Forum: The Human Cell Atlas". In: *eLife* 6 (Dec. 2017). Ed. by Thomas R Gingeras, e27041. ISSN: 2050-084X. DOI: 10.7554/eLife.27041. URL: https://doi.org/10.7554/eLife.27041.

[23] Haotian Cui et al. "scGPT: Towards Building a Foundation Model for Single-Cell Multi-omics Using Generative AI". In: *bioRxiv* (2023). DOI: 10.1101/2023.04.30.538439. eprint: https://www.biorxiv.org/content/early/2023/05/01/2023.04.30.538439.full.pdf. URL: https://www.biorxiv.org/content/early/2023/05/01/2023.04.30.538439.

[24] Minsheng Hao et al. "Large Scale Foundation Model on Single-cell Transcriptomics". In: *bioRxiv* (2023). DOI: 10.1101/2023.05.29.542705. eprint: https://www.biorxiv.org/content/early/2023/06/02/2023.05.29.542705.full.pdf. URL: https://www.biorxiv.org/content/early/2023/06/02/2023.05.29.542705.

[25] Christina V Theodoris et al. "Geneformer: Transfer learning enables predictions in network biology". In: *Nature* (2023), pp. 1–9.

[26] Alec Radford et al. "Language models are unsupervised multitask learners". In: *OpenAI blog* 1.8 (2019), p. 9.

[27] Brian Lester, Rami Al-Rfou, and Noah Constant. "The Power of Scale for Parameter-Efficient Prompt Tuning". In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 3045–3059. DOI: 10.18653/v1/2021.emnlp-main.243. URL: https://aclanthology.org/2021.emnlp-main.243.

[28] Tianyu Gao, Adam Fisch, and Danqi Chen. "Making Pre-trained Language Models Better Few-shot Learners". In: *Association for Computational Linguistics (ACL)*. 2021.

[29] Xiang Lisa Li and Percy Liang. *Prefix-Tuning: Optimizing Continuous Prompts for Generation*. 2021. arXiv: 2101.00190 [cs.CL].

[30] Rohan Taori et al. *Stanford Alpaca: An Instruction-following LLaMA model*. https://github.com/tatsu-lab/stanford_alpaca. 2023.

[31] Jason Wei et al. "Finetuned Language Models are Zero-Shot Learners". In: *International Conference on Learning Representations*.

[32] Shayne Longpre et al. "The Flan Collection: Designing Data and Methods for Effective Instruction Tuning". In: *arXiv preprint arXiv:2301.13688* (2023).

[33] Hyung Won Chung et al. *Scaling Instruction-Finetuned Language Models*. 2022. eprint: 2210.11416.

[34] Edward J Hu et al. "LoRA: Low-Rank Adaptation of Large Language Models". In: *International Conference on Learning Representations*. 2022. URL: https://openreview.net/forum?id=nZeVKeeFYf9.

[35] Pengfei Liu et al. "Pre-Train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing". In: *ACM Comput. Surv.* 55.9 (Jan. 2023). ISSN: 0360-0300. DOI: 10.1145/3560815. URL: https://doi.org/10.1145/3560815.

[36] Tadas Baltrušaitis, Chaitanya Ahuja, and Louis-Philippe Morency. "Multimodal machine learning: A survey and taxonomy". In: *IEEE transactions on pattern analysis and machine intelligence* 41.2 (2018), pp. 423–443.

[37] Arnab Barua, Mobyen Uddin Ahmed, and Shahina Begum. "A Systematic Literature Review on Multimodal Machine Learning: Applications, Challenges, Gaps and Future Directions". In: *IEEE Access* 11 (2023), pp. 14804–14831. DOI: 10.1109/ACCESS.2023.3243854.

[38] Lei Wu et al. "Visual language modeling for image classification". In: *Proceedings of the international workshop on Workshop on multimedia information retrieval*. 2007, pp. 115–124.

[39] C Domınguez Conde et al. "Cross-tissue immune cell analysis reveals tissue-specific features in humans". In: *Science* 376.6594 (2022), eabl5197.

[40] Haotian Cui et al. "scGPT: Towards Building a Foundation Model for Single-Cell Multi-omics Using Generative AI". In: *bioRxiv* (2023), pp. 2023–04.

[41] Wenpin Hou and Zhicheng Ji. "Reference-free and cost-effective automated cell type annotation with GPT-4 in single-cell RNA-seq analysis". In: *bioRxiv* (2023), pp. 2023–04.

[42] Fan Yang et al. "scBERT as a large-scale pretrained deep language model for cell type annotation of single-cell RNA-seq data". In: *Nature Machine Intelligence* 4.10 (2022), pp. 852–866.

[43] Chikara Furusawa and Kunihiko Kaneko. "Zipf's law in gene expression". In: *Physical review letters* 90.8 (2003), p. 088102.

[44] Xing Qiu, Hulin Wu, and Rui Hu. "The impact of quantile and rank normalization procedures on the testing power of gene differential expression analysis". In: *BMC bioinformatics* 14 (2013), pp. 1–10.

[45] Romain Lopez et al. "Deep generative modeling for single-cell transcriptomics". In: *Nature methods* 15.12 (2018), pp. 1053–1058.

[46] Arthur Gretton et al. "A Kernel Two-Sample Test". In: *Journal of Machine Learning Research* 13.25 (2012), pp. 723–773. URL: http://jmlr.org/papers/v13/gretton12a.html.

[47] Jacob C Kimmel, David G Hendrickson, and David R Kelley. "Differentiation reveals the plasticity of age-related change in murine muscle progenitors". In: *bioRxiv* (2020). DOI: 10.1101/2020.03.05.979112. eprint: https://www.biorxiv.org/content/early/2020/03/06/2020.03.05.979112.full.pdf. URL: https://www.biorxiv.org/content/early/2020/03/06/2020.03.05.979112.

[48] F Alexander Wolf, Philipp Angerer, and Fabian J Theis. "SCANPY: large-scale single-cell gene expression data analysis". In: *Genome biology* 19 (2018), pp. 1–5.

[49] Ashraful Haque et al. "A practical guide to single-cell RNA-sequencing for biomedical research and clinical applications". In: *Genome medicine* 9.1 (2017), pp. 1–12.

[50] Ilya Loshchilov and Frank Hutter. "Decoupled weight decay regularization". In: *arXiv preprint arXiv:1711.05101* (2017).

[51] Rico Sennrich, Barry Haddow, and Alexandra Birch. "Neural machine translation of rare words with subword units". In: *arXiv preprint arXiv:1508.07909* (2015).

[52] HF Canonical Model Maintainers. *gpt2 (Revision 909a290)*. 2022. DOI: 10.57967/hf/0039. URL: https://huggingface.co/gpt2.

[53] Tri Dao et al. "FlashAttention: Fast and Memory-Efficient Exact Attention with IO-Awareness". In: *Advances in Neural Information Processing Systems*. 2022.

[54] Tri Dao. "FlashAttention-2: Faster Attention with Better Parallelism and Work Partitioning". In: (2023).

Appendix

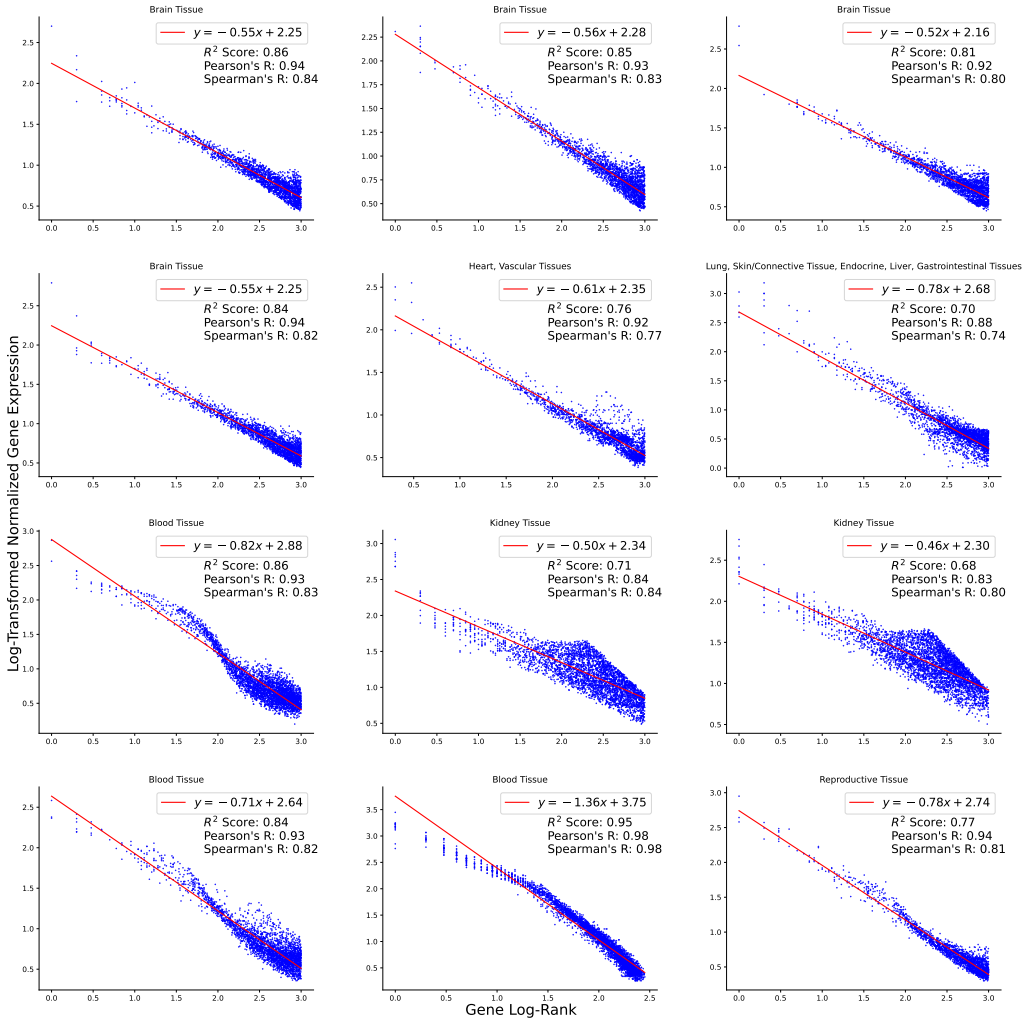

Figure 7: Scatter plots showing the log-linear relationship between gene expression and rank across 12 diverse scRNA-seq datasets. The red line shows the fitted linear model, with Pearson's R, Spearman's R, and R^2 quantifying goodness of fit. The high correlation values demonstrate that gene rank encodes expression in a consistent, reversible way across datasets. This enables translating between the text domain of cell sentences and original gene expression.

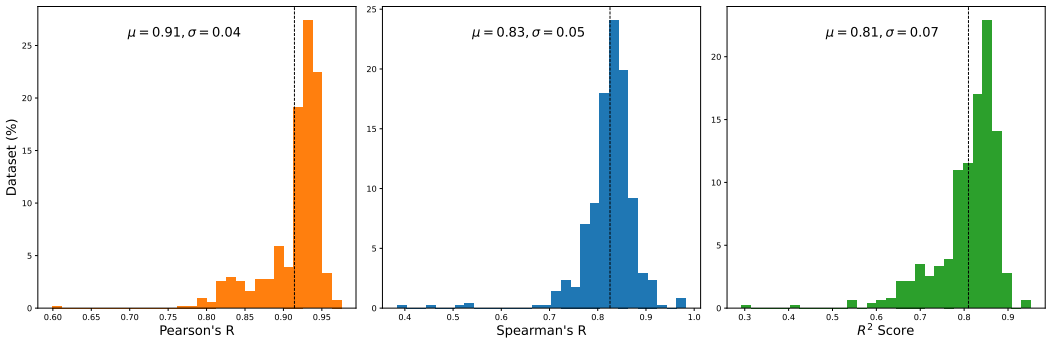

Figure 8: Distributions of Peason's R, Spearman's R and R^2 statistics computed on the performance of expression reconstruction from log-rank using a linear regression across 127 distinct scRNA-seq datasets. Both mean and standard deviation are shown for each score, with the mean represented by a dashed line on the x-axis.

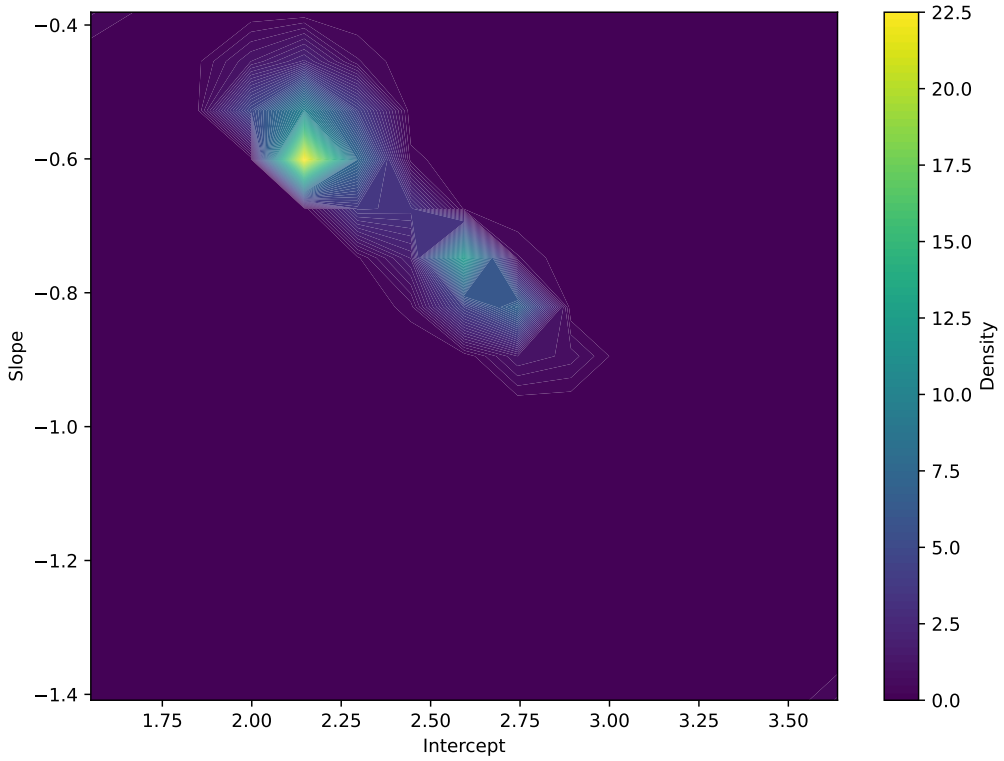

Figure 9: Density plots of fitted linear regression coefficients relating gene expression to rank across 127 scRNA-seq datasets. The coefficients are used to reconstruct expression from rank, with performance shown in Figure 8. The variability in optimal coefficients highlights the ability to customize reconstruction for diverse datasets.

Cell Type	GPT-2 Small (C2S)			GPT-2 Small (NL + C2S)			GPT-2 Medium (C2S)			GPT-2 Medium (NL + C2S)		
	PR	SR	R^2	PR	SR	R^2	PR	SR	R^2	PR	SR	R^2
CD16-, CD56-bright nk cell	0.964	0.730	0.928	**0.991**	0.733	**0.978**	0.967	0.755	0.930	0.990	**0.762**	0.972
CD16+, CD56-dim nk cell	0.951	0.773	0.900	0.989	0.765	0.973	0.946	0.773	0.892	**0.991**	**0.802**	**0.976**
CD4+ helper T cell	0.948	0.791	0.894	0.984	0.790	**0.957**	0.945	0.788	0.888	**0.984**	**0.815**	0.955
CD8+, α-β memory T cell	0.952	0.790	0.899	**0.989**	0.827	**0.971**	0.952	0.812	0.901	0.987	**0.835**	0.965
CD8+, α-β memory T cell, CD45RO+	0.982	0.816	0.953	0.992	0.796	**0.975**	0.972	0.809	0.936	**0.993**	**0.828**	0.974
T follicular helper cell	0.982	**0.810**	**0.944**	0.983	0.776	0.933	**0.983**	0.799	0.941	0.975	0.769	0.903
classical monocyte	0.767	0.653	0.549	0.979	0.729	0.937	0.802	0.713	0.630	**0.982**	**0.769**	0.941
effector memory CD4+, α-β T cell	0.985	0.875	0.926	0.990	0.853	0.944	0.987	0.895	0.927	0.989	0.867	**0.946**
effector memory CD8+, α-β T cell	0.967	**0.782**	0.923	0.990	0.765	0.972	0.961	0.768	0.914	**0.992**	0.759	**0.976**
gamma-delta T cell	0.976	0.769	0.945	**0.984**	0.793	**0.963**	0.942	0.770	0.885	0.982	**0.807**	0.958
macrophage	0.719	0.576	0.347	0.938	0.715	0.877	0.747	0.669	0.443	**0.940**	**0.759**	**0.881**
memory B cell	**0.986**	0.748	**0.956**	0.984	0.710	0.928	**0.986**	**0.757**	0.954	**0.986**	0.689	0.935
mucosal invariant T cell	0.981	0.761	0.954	**0.988**	**0.763**	**0.962**	0.976	0.757	0.945	0.985	0.745	0.948
naive B cell	0.987	0.794	**0.959**	**0.989**	0.764	**0.959**	0.986	**0.803**	0.955	0.988	0.752	0.948
naive thymus-derived CD4+, α-β T cell	0.982	0.825	0.907	**0.993**	0.805	**0.952**	0.988	**0.837**	0.930	**0.993**	0.815	**0.952**
naive thymus-derived CD8+, α-β T cell	0.984	0.817	0.904	**0.994**	0.808	**0.952**	0.988	**0.860**	0.933	0.993	0.818	0.949
regulatory T cell	**0.985**	0.716	**0.958**	0.966	0.670	0.898	0.983	**0.717**	0.956	0.948	0.686	0.849

Table 6: Quantification of cell generation accuracy per cell type. PR (Pearson correlation coefficient), SR (Spearman's rank correlation coefficient), and R^2 (coefficient of determination) metrics compare averaged generated cells versus averaged ground truth cells. The high correlation values across multiple cell types demonstrate Cell2Sentence can accurately generate diverse cell types. Pretraining on natural language (NL) further improves performance.

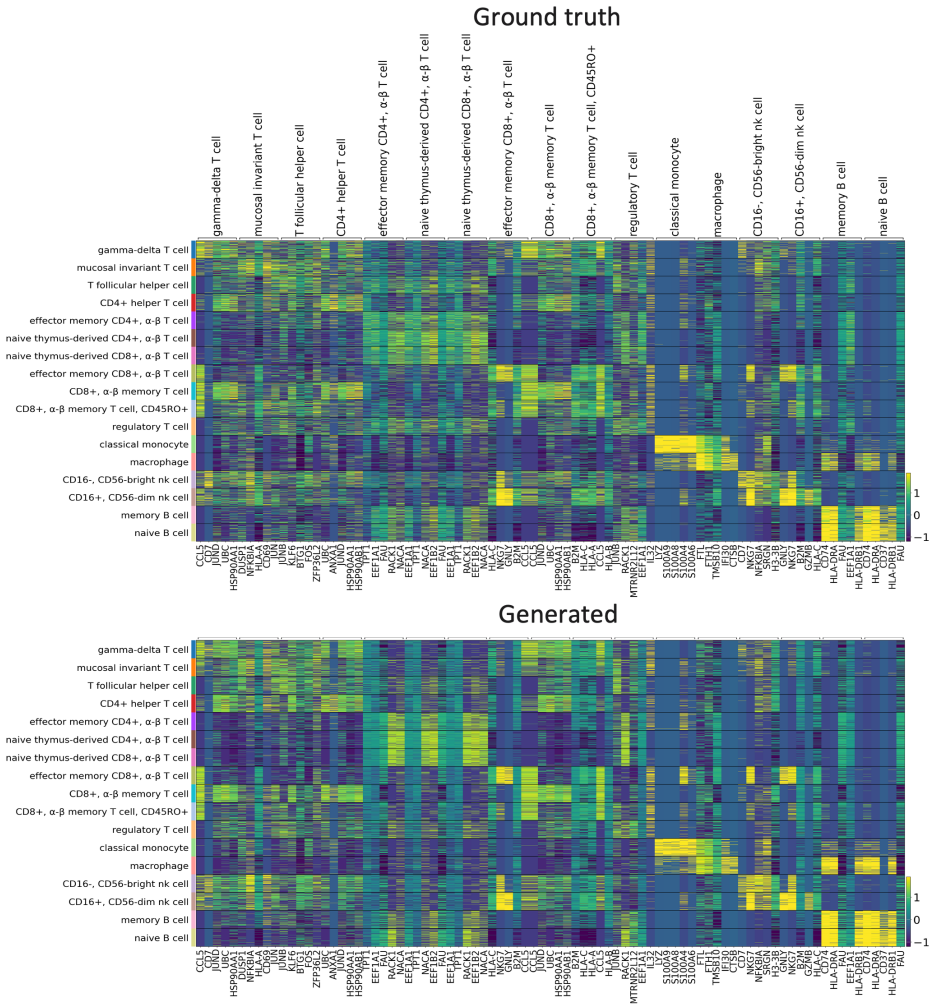

Figure 10: Heatmap visualizations comparing differentially expressed genes per cell type between generated and ground truth expression data showing strong correspondence between generated and real cells across a number of cell types. Differentially expressed genes were identified from the top 100 genes in ground truth data.

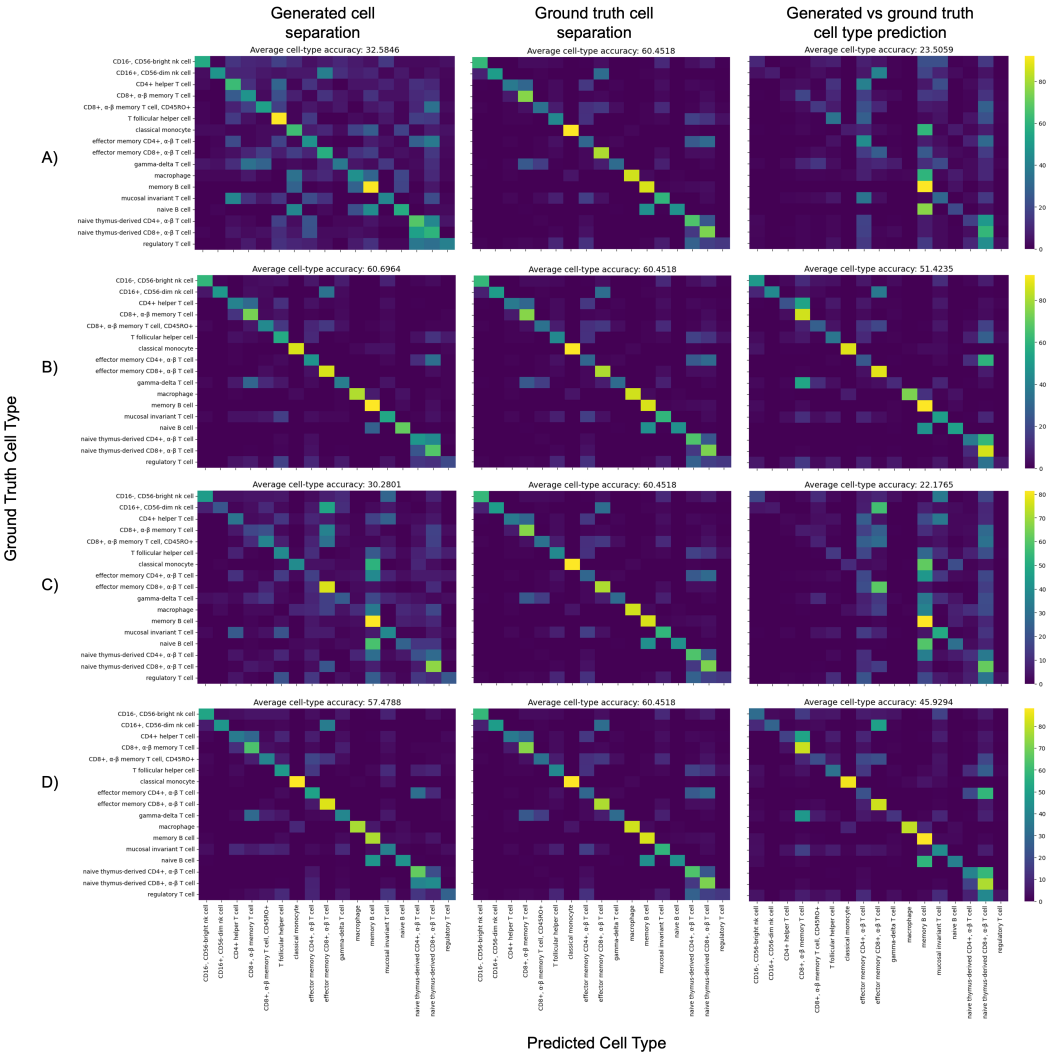

Figure 11: Confusion matrices for KNN classification of cell types using 25 nearest neighbors. Models are (A) GPT-2 Small pretrained, (B) GPT-2 Small fine-tuned, (C) GPT-2 Medium pretrained, and (D) GPT-2 Medium fine-tuned. Fine-tuned models show clearer diagonal patterns, indicating better separation of cell types. This demonstrates improved generation fidelity from natural language pretraining and model scaling.

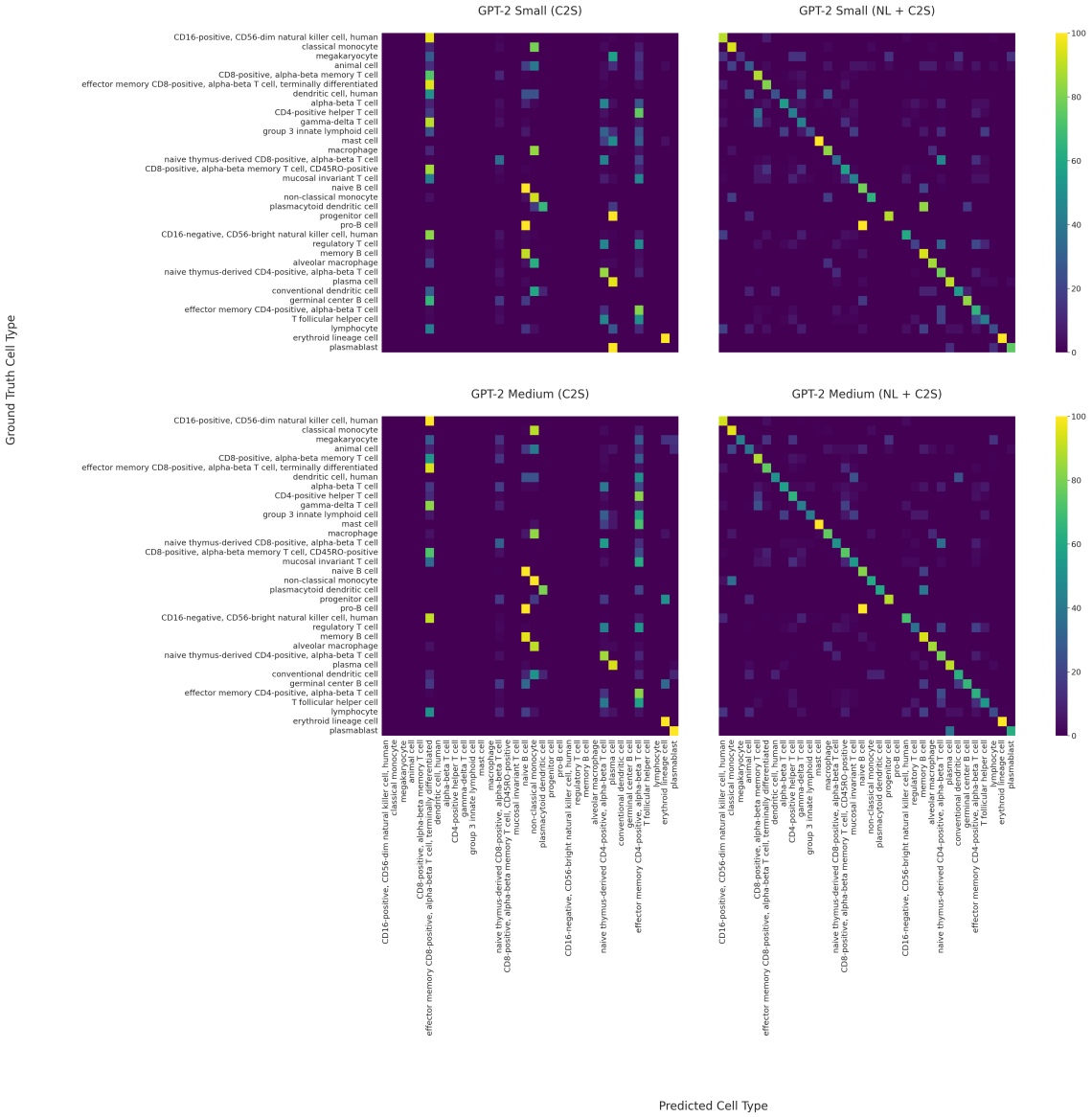

Figure 12: Confusion matrices for autoregressive cell type prediction on unseen immune tissue test data. Models pretrained on natural language text before Cell2Sentence training (NL + C2S) significantly outperform models trained only on cell sentences (C2S). This demonstrates the benefits of transfer learning from natural language for predicting cell types from gene sequences.

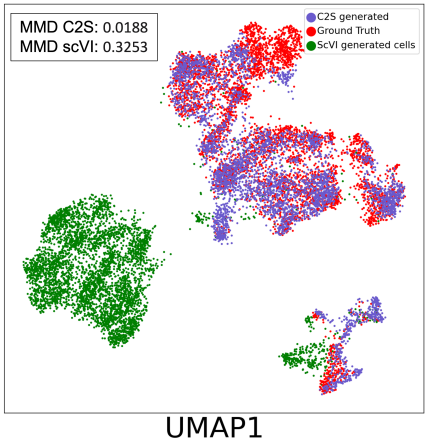

Figure 13: UMAP plots comparing cell embeddings from C2S, scVI, and ground truth data. The Maximum Mean Discrepancy (MMD) statistic quantifies the difference in distributions. The lower MMD value for C2S indicates that our generated cells better match the real data distribution than the cells generated with scVI.

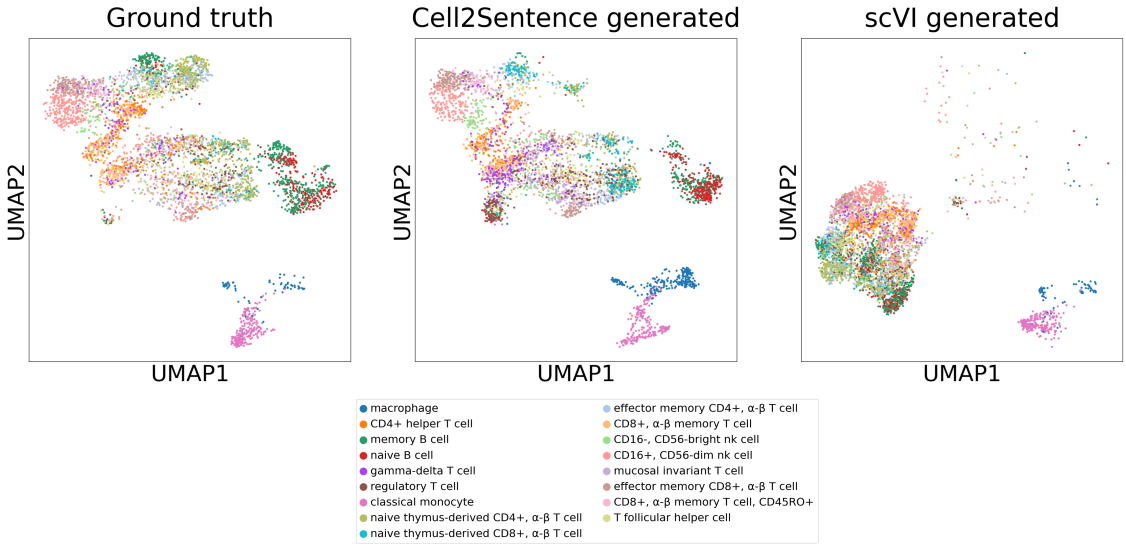

Figure 14: UMAP visualization colored by cell type, comparing C2S, scVI, and ground truth data. Both C2S and scVI generate distinct cell types, but only C2S cells strongly align with ground truth cells of the matching type. This shows that the C2S framework can generate specific cell types that closely mimic real cells, outperforming scVI.