EFFICIENT DISTRIBUTIONAL ENCODING FOR MULTI-CELLULAR RNA-SEQ DATA

Anonymous authors
Paper under double-blind review

ABSTRACT

Single-cell RNA-seq models typically tokenize cells independently as sequences of gene or cell-state tokens, causing sequence length to scale linearly with number of cells $N\times$ genes per cell G. This scaling is poorly suited to tasks requiring multi-cellular context, such as patient-level disease diagnosis and spatial microenvironment niche mapping, where samples comprise dozens to thousands of cells. Here, we present a benchmarking study of single-cell tokenization strategies for multi-cellular prediction tasks. Holding a standard transformer backbone fixed, we evaluate gene- and cell-level tokenizers across three multi-cellular benchmarks, tracking predictive performance and token usage as the cellular cohort size varies from 1 to 1,000 cells. To overcome linear context expansion, we introduce module distribution encoding (MDE). MDE groups correlated genes into co-expression modules and summarizes the population-level expression profile of each module as a zero-adjusted distribution token. Total sequence length is therefore determined strictly by the number of gene modules M rather than cell or gene count, decoupling context length from cohort size while preserving continuous population-level variance. MDE outperforms existing tokenization schemes on disease prediction and niche prediction, and cuts tokens by orders of magnitude, leading to significantly higher efficiency across all tasks. These results characterize the context-accuracy trade-offs of single-cell tokenization and establish sublinear, population-aware encodings as an effective basis for multi-cellular modeling.

1 INTRODUCTION

Advancements in single-cell RNA sequencing (scRNA-seq) technologies have driven the growth of massive transcriptomic atlases spanning millions of cells across tissues, donors, and disease conditions (Dai & Shen, 2022; Regev et al., 2017; Program et al., 2024). These atlases have motivated the development of transcriptomic foundation models that train across diverse scRNA-seq data to learn generalizable representations of genes and cell states (Rizvi et al., 2026; Theodoris et al., 2023; Hao et al., 2024; Rosen et al., 2023). Existing architectures have demonstrated predictive capabilities on cell-level benchmarks, including cell-type annotation, batch integration, gene function inference, and in-silico perturbation effect modeling (Rizvi et al., 2026; Theodoris et al., 2023; Adduri et al., 2025). However, many biological and clinical phenotypes are properties of multi-cellular contexts rather than isolated cells. Tasks such as patient-level disease diagnosis and spatial niche characterization require reasoning over populations ranging from dozens to thousands of cells per sample. Extending standard transformer architectures to these multi-cellular regimes remains fundamentally constrained by tokenization: current token schemes scale compute and context requirements with the number of input cells, impeding their utility for sample-level representation learning.

The primary bottleneck in applying current transcriptomic foundation models to multi-cellular contexts lies in their tokenization strategies. Existing models tokenize individual cells at the gene level, encoding each gene as a distinct token with an identity and value encoding, and relying on subsampling nonzero expressed genes or restricting to highly variable genes (HVGs) to bound context length (Cui et al., 2024; Theodoris et al., 2023; Hao et al., 2024). For a multi-cellular cohort of N cells, sequence length under this formulation scales as $\mathcal{O}(N \cdot G)$, where G denotes the per-cell gene tokens (Figure 1b). Consequently, self-attention memory overhead ($\mathcal{O}((N \cdot G)^2)$) rapidly be-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

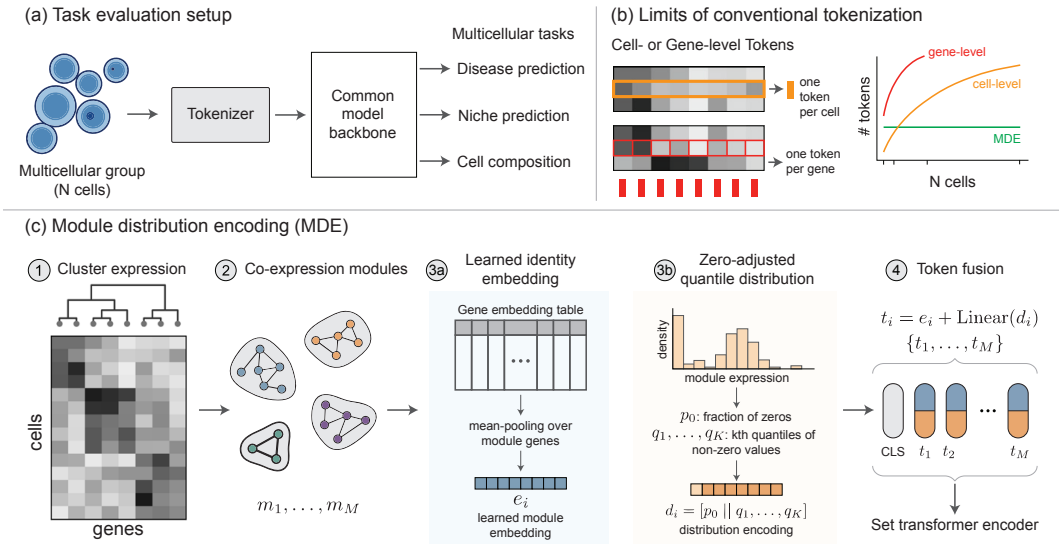


Figure 1: Overview of the tokenization study and module distribution encoding. **(a)** Multi-cellular cohorts are tokenized with standard single-cell schemes and with MDE , then trained on downstream tasks using the same transformer backbone to isolate differences in performance and token use to the tokenizer only. **(b)** Existing tokenization schemes scale poorly with cohort size: gene-level tokenization grows as $\mathcal{O}(N \cdot G)$, while cell projection grows linearly as $\mathcal{O}(N)$. **(c)** MDE instead aggregates co-expressed genes into modules and encodes each module's expression distribution across the cohort as a single token, fixing the representation cost of one feature for any cohort size N. Sequence length is therefore set by the (constant) module budget M rather than by N or G.

comes intractable for transformer-based models even for modest cell cohort sizes. Recent multi-cell architectures such as STATE (Adduri et al., 2025) and STACK (Dong et al., 2026) mitigate this by mapping each cell to a continuous embedding vector, thereby reducing sequence length to $\mathcal{O}(N)$ tokens for a cohort of N cells. While this is better than gene-level tokenization, linear scaling of tokens relative to cohort size remains computationally expensive for clinical cohorts comprising hundreds to thousands of cells. Moreover, STACK's gene-module tokens are still computed independently within each cell, and neither STATE nor STACK summarizes shared expression distributions across the cohort, treating homogeneous cell ensembles as long, redundant sequences rather than as a population-level transcriptional state.

In this work, we present a benchmarking study of existing tokenization strategies for multi-cellular prediction tasks. To decouple the effect of the tokenization method from the model architecture, we use the same transformer backbone across models, and evaluate gene-level, cell-level, and population-level tokenization frameworks across three multi-cellular tasks: patient disease classification, spatial niche prediction, and cell composition estimation. We measure performance and token cost of each tokenization framework as the cellular cohort size (N) varies for each task. To address the linear token scaling of existing methods, we introduce module distribution encoding (MDE): rather than representing cells or genes as independent tokens, MDE groups correlated genes into biological co-expression modules and summarizes the population-level expression distribution of each module as a compact, zero-adjusted quantile distribution token. Consequently, sequence length is determined strictly by the chosen module resolution M rather than cohort size N or gene count G, decoupling token budget from the number of input cells while preserving continuous population-level variance. Across evaluated benchmarks, MDE outperforms gene- and cell-level tokenizers on disease and niche prediction while cutting token requirements by orders of magnitude—and thus improving efficiency—across all tasks, demonstrating that distribution-aware tokenization provides an efficient, sublinear foundation for multi-cellular transformer modeling.

2 Related Works

2.1 Transcriptomic Foundation Models

Recent years have seen many proposed transcriptomic foundation models trained on large-scale curated datasets of publicly-available scRNA-seq data. These datasets typically come from single-cell databases such as Human Cell Atlas (Regev et al., 2017) and CELLxGENE (Program et al., 2024). Through large-scale self-supervised learning, these foundation models learn latent representations of cells and genes, and are applied to downstream tasks such as cell-type annotation, perturbation prediction, batch correction, and gene function inference (Cui et al., 2024; Theodoris et al., 2023; Hao et al., 2024; Rosen et al., 2023; Adduri et al., 2025; Yang et al., 2022).

Gene-level tokenization Single-cell foundation models employ different tokenization strategies that center around encoding genes as tokens. scGPT (Cui et al., 2024) learns a vocabulary of gene tokens to identify genes, and performs adaptive binning of expression values per gene into 51 discrete bins to convert exact expression values into relative binned values per cell. The gene identity embedding and binned value embedding are added together to get the final representation for each gene in a given cell. Geneformer (Theodoris et al., 2023) similarly learns a vocabulary of gene tokens; however, it normalizes each gene's expression across the entire dataset, and rank orders genes within a given cell to form a non-parametric representation of the transcriptome of each single cell. Under this rank ordering, genes that distinguish a cell's state relative to all other cells will be prioritized in the rank ordering, appearing earlier in the sentence. Cell2Sentence and C2S-Scale (Levine et al., 2023; Rizvi et al., 2026) extend this idea by forming a rank-ordered natural language "cell sentence" of gene names to represent each cell for large language models (LLMs). Other related gene-token models include scBERT (Yang et al., 2022), Universal Cell Embeddings (UCE) (Rosen et al., 2023), and scFoundation (Hao et al., 2024), which likewise represent each cell as a sequence or embedding derived from per-gene inputs. All of these tokenization frameworks scale the number of resulting tokens with the number of cells times the number of genes.

Cell-level tokenization Recent multi-cell foundation models for single-cell transcriptomics have been developed for tasks such as perturbation prediction. These models typically represent each cell as one token, and use a sequence of cell tokens to encode multi-cellular context. STATE (Adduri et al., 2025) excels on single-cell perturbation over sets of cells. STACK (Dong et al., 2026) projects each cell into a group of "gene module" token vectors, decomposing a cell into a collection of about 100 tokens to group biologically relevant genes together. Similar to MDE , STACK's formulation effectively learns a fixed gene module aggregation; however, it is static and occurs independently within each cell, with no distributional encoding across cells. STACK and STATE's tokenization schemes therefore scale with the number of cells, which remain expensive for substantially large multi-cell groupings. CellFlow (Klein et al., 2025) established a flow-matching framework over sets of cell populations for perturbation effect prediction, using principal component analysis (PCA) and variational autoencoders (VAEs) (Kingma & Welling, 2013) to project cell transcriptomes into lower dimensional representations. Relatedly, scSLIDE (Jiang et al., 2025) builds sample-level embeddings by projecting cells into a joint cell-type and disease-aware latent space and summarizing each patient sample by densities around landmark cells; unlike MDE , the representation is a density over cell embeddings rather than expression distributions over gene modules.

2.2 Patient-Level and Multi-Cellular Modeling

Outside foundation-model tokenizers, multi-cellular phenotypes are typically handled by aggregating cells into sample-level summaries. Pseudobulk methods collapse expression within a sample (often by cell type) for bulk-style analysis and are standard for avoiding cell-level pseudoreplication (Crowell et al., 2020; Squair et al., 2021). Supervised alternatives cast each donor as a bag of cells and pool embeddings with attention or related set aggregators (He et al., 2021a; Mao et al., 2024; Liu et al., 2026), while spatial niche methods summarize local neighborhoods for microenvironment characterization (Qian et al., 2025; Birk et al., 2025). These approaches predict sample labels effectively, but reduce cohorts to means, pooled embeddings, or niche features, limiting their expressivity.

3 METHODOLOGY

We next define module distribution encoding (MDE ; Figure 1c). The shared transformer backbone, baseline tokenizations, tasks, and evaluation protocol are described in Section 4.

3.1 MODULE DISTRIBUTION ENCODING

Module distribution encoding (MDE) represents a multi-cellular cohort by summarizing population-level expression of co-expression gene modules, rather than tokenizing cells or genes independently. Sequence length is therefore set by the module budget M and is independent of cohort size N.

Gene module aggregation. On the training split, we compute a gene–gene Pearson correlation matrix (typically over highly variable genes), form the correlation distance $D = 1 - R$, and apply hierarchical clustering with Ward linkage, cutting the dendrogram into M modules. For each cell n and module i, the module score $x_{n,i}$ is the mean expression of genes assigned to that module. Larger M increases feature resolution at the cost of more tokens; we use $M{=}100$ for most experiments ($M{=}50$ for spatial niche prediction). The choice of M is studied in Section 4.4.

Zero-adjusted quantile encoding. For each module i, we summarize the empirical distribution of $\{x_{n,i}\}_{n=1}^{N}$ with a fixed-length vector of dimension K (default $K{=}50$). Let $p_{0,i}$ denote the fraction of cells with (near-)zero module score. The remaining $K{-}1$ coordinates are evenly spaced empirical percentiles of the *nonzero* scores:

$$p_{0,i} = \frac{1}{N}\sum_{n=1}^{N} \mathbf{1}\{|x_{n,i}| \le \varepsilon\}, \tag{1}$$

$$q_{i,k} = F_{i,+}^{-1}(\tau_k)\,, \qquad \tau_k = \frac{k}{K-1}, \quad k = 1, \ldots, K-1, \tag{2}$$

where $F_{i,+}^{-1}$ is the empirical quantile function on $\{x_{n,i} : |x_{n,i}| > \varepsilon\}$. If all scores are zero, we set $p_{0,i}{=}1$ and $q_{i,k}{=}0$. The zero-adjusted distribution vector is

$$d_i = \left[p_{0,i},\, q_{i,1}, \ldots, q_{i,P-1}\right] \in \mathbb{R}^K. \tag{3}$$

The effect of this zero-adjustment is studied in Section 4.5. A linear projection maps each d_i to the model hidden dimension, yielding one token per module and decoupling token cost from N.

Token construction. We learn a gene vocabulary embedding and form the module identity e_i by averaging the embeddings of genes in module i. The final module token is

$$t_i = e_i + \mathrm{Linear}(d_i). \tag{4}$$

We note that gene identity embeddings can in principle be recombined into alternative module definitions (e.g., Gene Ontology terms (Ashburner et al., 2000)); we leave foundation-model pretraining and variable module vocabularies to future work, and focus here on the tokenization benefits of gene aggregation.

4 EXPERIMENTS

4.1 SHARED TRANSFORMER ARCHITECTURE

To isolate the effect of tokenization, all sequence-based methods share a 4-layer transformer backbone with a linear prediction head on the final-layer `[CLS]` token. Disease prediction and cell composition use hidden size $d_{\text{model}}{=}128$, 8 attention heads, and feed-forward width $d_{\text{ff}}{=}256$; spatial niche prediction uses $d_{\text{model}}{=}256$ and $d_{\text{ff}}{=}512$. We compare four tokenization schemes under this backbone: scGPT-style and Geneformer-style gene-level encodings, cell projection, and MDE . Geneformer rank-ordering and scGPT value binning with gene identity embeddings are re-implemented from the authors' releases; cell projection maps each cell to one continuous token via a single linear layer. For MDE , positional encodings are disabled so module tokens are treated as an unordered set. Full training hyperparameters are reported in Appendix B. Alongside task metrics, we count non-padding tokens immediately after tokenization and before the backbone, excluding the shared `[CLS]` token.

4.2 Evaluation Tasks

Disease Prediction We evaluate patient-level disease classification on the Guimaraes et al. myeloid dataset (Guimarães et al., 2024) (∼215k cells, 51 donors, 15 disease labels). Because many diseases appear in only one donor, we split cells 80/20 within each donor–disease pair and sample multi-cell cohorts from a single donor and disease condition; preprocessing and sample construction are detailed in Appendix A.1. The model receives N cells and predicts one of 15 disease classes via categorical cross-entropy on the `[CLS]` representation. We sweep $N \in \{1, 3, 5, 10, 20, 50\}$. For MDE we use $M{=}100$ co-expression modules and $K{=}50$ zero-adjusted quantile dimensions; gene-level baselines use up to 1,000 gene tokens per cell. We report balanced accuracy as the primary metric, with balanced accuracy and weighted-F1 in the appendix.

Spatial Niche Prediction Spatial niche prediction is a local microenvironment composition task on the CosMx human liver dataset (He et al., 2021b). We use 306,524 cells from healthy tissue with expression over 1,000 curated genes and six annotated niches (Central Vein, Portal, Zone 1, Zone 2a, Zone 2b, Zone 3a). For each sample, an anchor cell and its $N{-}1$ spatial nearest neighbors define a cohort. The target is the empirical niche proportion vector over those N cells (Appendix A.3). We evaluate $N \in \{1, 5, 10, 20, 50, 100, 200, 500\}$. MDE uses $M{=}50$ modules and $K{=}50$; gene-level baselines are budgeted at up to 300 gene tokens per cell, which captures 99% of expressed genes across all cells in the dataset. Models are trained with soft cross-entropy loss. We report mean per-sample Spearman correlation of niche proportions as the primary metric, with Jensen-Shannon divergence and top-1 accuracy in the appendix.

Cell Composition Prediction Immune cell composition estimation uses the Dominguez et al. cross-tissue dataset (Domínguez Conde et al., 2022) (∼330k cells, 12 donors). We restrict targets to the 28 cell subtypes shared across donors and use a donor-level train/val/test split. Multi-cell cohorts of size N are sampled to match controlled composition targets with sparsity locking so that target sparsity is comparable across N (Appendix A.2). The model predicts the 28-dimensional cell-type proportion vector for each cohort. We evaluate $N \in \{20, 50, 100, 200, 500, 1000\}$. MDE uses $M{=}100$ and $P{=}50$; gene-level baselines use up to 1,000 gene tokens per cell. Training minimizes soft cross-entropy; primary reported metrics are the Spearman correlation across the most prevalent 15 cell types in each sample and the Jensen–Shannon divergence.

4.3 Results

We evaluated MDE against standard single-cell tokenization strategies across three multi-cellular tasks: disease classification, spatial niche composition prediction, and 28-subtype immune cell composition estimation. Across all benchmarks, task performance improves monotonically as the multi-cellular context (N) expands, confirming that increased cellular context provides additional discriminative signal compared to individual cells for these tasks. In disease prediction, MDE achieves superior weighted accuracy over all baselines across every cohort size $N \geq 5$, reaching 0.99 accuracy at $N = 20$ compared to 0.93 for cell projection and Geneformer (Figure 2 top row). In spatial niche prediction, MDE performs comparably to cell projection across all neighborhood sizes, with both methods rising from ~ 0.45 Spearman correlation at $N = 1$ to approximately 0.90 at $N = 500$. In immune cell composition estimation, cell projection scales smoothly from a top-15 Spearman correlation of 0.78 at $N = 20$ to 0.90 at $N = 500$, while Geneformer and scGPT achieve substantially lower correlations (0.61 and 0.52, respectively) at $N = 20$. MDE outperforms both scGPT and Geneformer even at $N = 1$, but only achieves 0.747 top-15 Spearman correlation at $N = 500$.

We suspect cell projection is particularly well matched to the cell composition target because representing each cell as one token turns composition into something close to a counting problem: the model can aggregate tokens whose embeddings look similar, whereas distribution-level encodings discard per-cell identity in favor of population summaries. These performance metrics suggest that tokenizer choice is task-dependent: cell-level tokens suit explicit composition estimation, while distribution tokens excel when the label depends on cohort-wide expression structure.

The performance gains of MDE are coupled with a significant reduction in token usage and therefore better scalability to large cellular contexts. scGPT and Geneformer scale linearly with both cells and genes, causing context lengths to rapidly exceed 10^4 tokens even at modest cohort sizes ($N = 20$; Figure 2 middle row), causing training and inference to become computationally prohibitive at larger

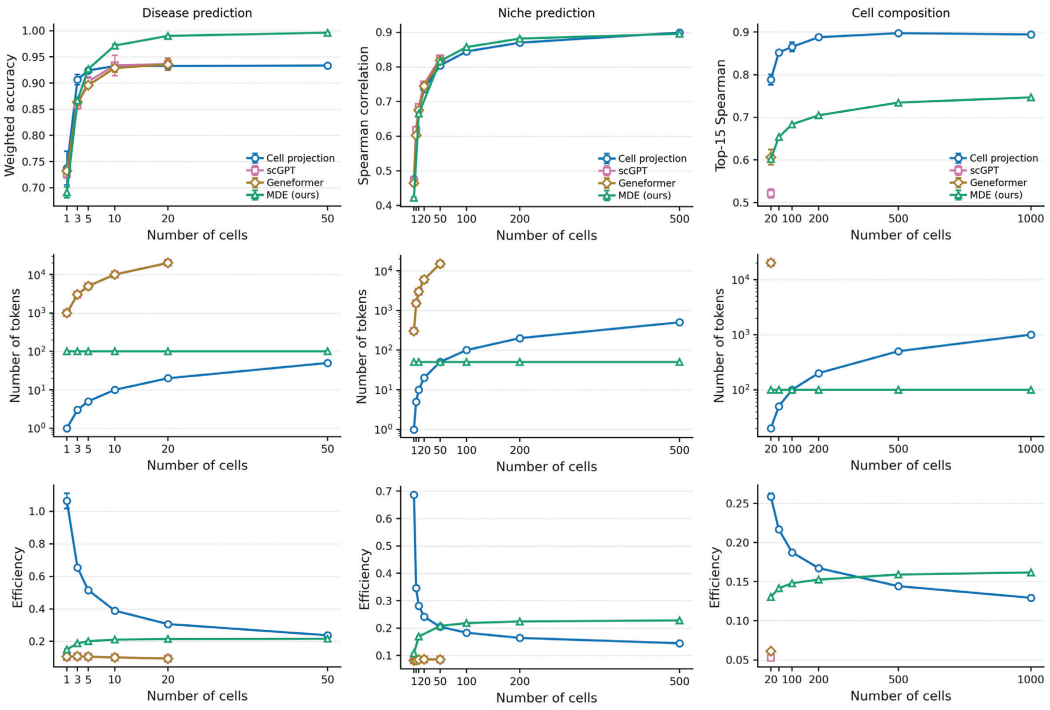


Figure 2: Benchmark evaluation of tokenization frameworks across multi-cellular prediction tasks and context scales. **Top row:** Predictive performance of each tokenization framework across varying number of input cells (N), shown for three predictive tasks: disease classification (evaluated by weighted accuracy on the myeloid dataset (Guimarães et al., 2024)), spatial microenvironment niche prediction (mean per-sample Spearman rank correlation of niche proportions on the CosMx liver dataset (He et al., 2021b)), and immune cell composition estimation (top-15 Spearman rank correlation of cell type proportions on the immune dataset (Domínguez Conde et al., 2022)). Across all settings, predictive accuracy scales with increasing cell cohort size before plateauing. **Middle row:** Corresponding token usage for each tokenization framework plotted on logarithmic axes as a function of number of input cells (N). While gene-level tokenization methods (scGPT, Geneformer) scale as $\mathcal{O}(N \cdot G)$ and cell projection scales linearly as $\mathcal{O}(N)$, MDE maintains a constant token sequence length invariant to the input number of cells, determined solely by the number of gene modules ($M = 100$ for disease and $M = 50$ for niche prediction). **Bottom row:** Downstream predictive efficiency, defined as task performance divided by log tokens used (metric$/\log(\text{tokens})$), plotted as a function of cell cohort size N. Gene-level tokenizers (Geneformer, scGPT) show flat, low efficiency owing to large context footprints. Cell projection efficiency degrades rapidly as cellular context size grows, whereas MDE maintains stable or increasing efficiency by keeping token context length invariant to cohort size. Points denote test-set evaluation means, and error bars indicate standard deviations across evaluation cohorts.

cell counts. Cell projection mitigates this by mapping each cell to a single continuous token, yet its token sequence length still scales directly with the cohort size N. In contrast, MDE fixes sequence length strictly to the chosen number of co-expression modules ($M = 100$ in disease classification and $M = 50$ in spatial niche prediction), completely decoupling context length from input cohort size. By compressing multi-cellular inputs into distribution tokens, MDE maintains constant token consumption across cell cohort scales without sacrificing predictive accuracy.

We additionally evaluate the trade-off between predictive performance and token usage for different tokenization frameworks through task efficiency, defined as downstream performance divided by the logarithm of token sequence length, as cohort size varies (Figure 2 bottom row). Because cell projection incurs linear token growth with cohort size N, its efficiency metric drops sharply as N expands, decreasing from over 1.0 at $N = 1$ to approximately 0.24 at $N = 50$ on disease prediction as additional cell tokens yield diminishing returns relative to sequence length expansion. Geneformer

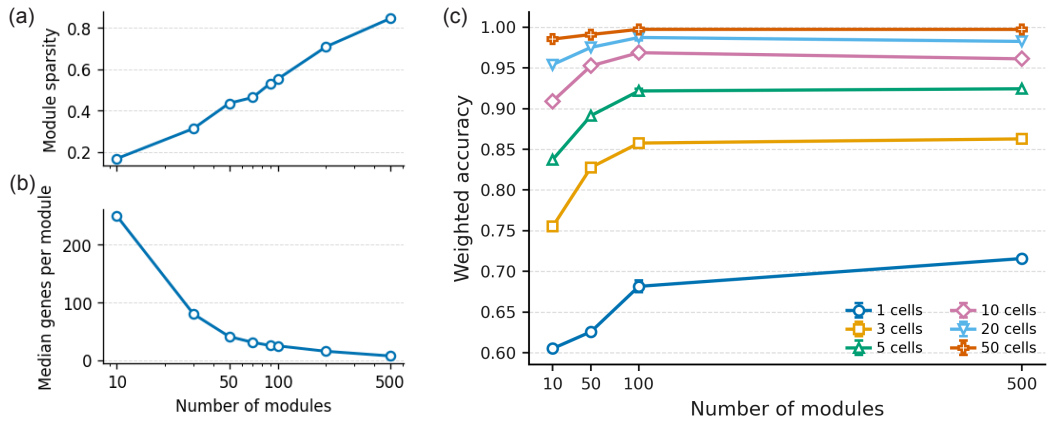


Figure 3: Effect of the module budget M on MDE for disease prediction. **(a)** Mean sparsity of module scores and **(b)** median genes per module as co-expression clustering is partitioned into $M \in \{10, 50, 100, 500\}$ modules. Finer budgets yield sparser, smaller modules. **(c)** Weighted accuracy versus M across cohort sizes $N \in \{1, 3, 5, 10, 20, 50\}$. Accuracy rises from M=10 to M=100 and then plateaus; M=500 adds little beyond M=100. Error bars denote standard deviation across seeds.

and scGPT exhibit consistently low efficiency (≤ 0.1) due to the large token footprint required by gene-level encodings. In contrast, MDE *increases* in efficiency across cohort sizes. Because MDE holds sequence length constant at M tokens regardless of N, the gains in task performance translate directly into sustained token efficiency even as more cellular context is encoded. Thus, while the preferred tokenizer for raw performance is task-dependent, MDE is the most token-efficient option across tasks, especially at large N.

4.4 Module Resolution

A central hyperparameter of MDE is the module budget M, which sets both feature resolution and sequence length. On the disease prediction task, we sweep $M \in \{10, 50, 100, 500\}$ (Figure 3). Increasing M produces finer co-expression modules: module-score sparsity rises and the median number of genes per module falls from hundreds of genes at M=10 to fewer than ten at M=500 (panels a–b). Predictive accuracy, however, saturates much earlier. Across cohort sizes, weighted accuracy improves sharply up to M=100 and then plateaus; expanding to M=500 yields only marginal gains (panel c). Larger cellular contexts N further lift accuracy at every M, so cohort size and module resolution act as complementary levers, but a moderate budget of M=100 already captures essentially all of the available signal for this task while keeping token cost fixed and modest.

4.5 Zero-Adjustment and Feature Aggregation

We next ablate two design choices in the distribution encoder on disease prediction (Figure 4). First, we compare zero-adjusted quantiles (nonzero percentiles plus an explicit zero fraction p_0) to the same number of percentiles computed over all values, including zeros (panel a). The two variants perform nearly identically for $N \geq 3$. At N=1, where module scores are sparsest, zero-adjustment is marginally better, consistent with dedicating capacity to zero inflation when few cells are observed. We therefore use zero-adjustment in all main experiments.

Second, we ask whether gene modules are necessary, or whether applying the same distribution encoding independently to every gene would suffice (panel b). Encoding all 21,843 genes or 5k HVGs as separate distribution tokens outperforms M=100 modules at N=1, where fine-grained single-gene signal dominates. From N=5 onward, however, the module variant overtakes both gene-level encodings and remains strongest through N=50, while using $50\times$ fewer tokens than 5k HVGs and $\approx 218\times$ fewer than the full gene set. Module aggregation is thus the ingredient that preserves multi-cellular accuracy under a fixed, compact token budget.

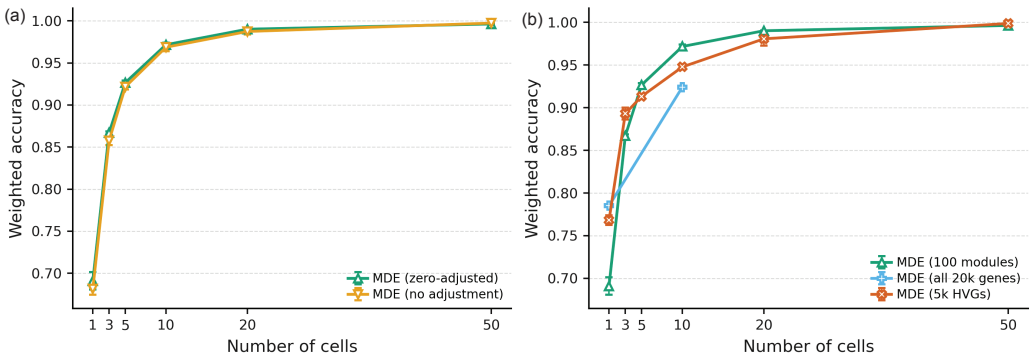


Figure 4: Design ablations of MDE on disease prediction. **(a)** Zero-adjusted quantiles versus quantiles over all values (including zeros). Curves are nearly overlapping for $N \geq 3$, with a small advantage for zero-adjustment at $N{=}1$. **(b)** Distribution tokens over $M{=}100$ co-expression modules versus over individual genes (5k HVGs or all 21,843 genes). Gene-level features lead at $N{=}1$, but modules overtake by $N{=}5$ and remain strongest thereafter at $50\times$–$\approx 218\times$ fewer tokens. Error bars denote standard deviation across seeds.

4.6 Identity and Distribution Geometry

Each MDE token combines a module identity embedding with a population-level distribution summary. To separate these two cues, we visualize identity and distribution geometry for disease-prediction models trained at $M{=}100$ (Figure 5). Joint PCA of the learned identity vectors e_i from models trained at $N \in \{1, 10, 50\}$, projected into a shared basis, yields essentially the same layout across cohort sizes (panel a). Modules of different membership sizes occupy stable positions, indicating that identity geometry is driven by gene membership rather than the training value of N.

In contrast, the zero-adjusted quantile encodings vary systematically with cohort size (panel b). For the same three modules in a fixed training sample, $N{=}1$ produces a step CDF at a single observed score, whereas $N{=}10$ and $N{=}50$ fill in smoother empirical distributions—including the zero-inflation mass captured by p_0. Thus identity provides a stable module address while the distribution token carries the N-dependent cellular context; summing them into t_i lets the transformer use both without increasing sequence length beyond M.

5 Conclusion

In this work, we presented a benchmarking study of single-cell tokenization strategies for multi-cellular prediction tasks and introduced module distribution encoding (MDE) to resolve the context scaling bottlenecks of existing transformer architectures. By organizing co-expressed genes into modules and summarizing cohort expression as zero-adjusted distribution tokens, MDE fixes input sequence length to the chosen module budget rather than the input number of cells. Across disease diagnosis, spatial niche mapping, and immune cell composition prediction, this distribution-level tokenization outperforms gene- and cell-level models on disease and niche prediction while operating at constant sequence length, and much lower token cost, across cohort sizes up to 1,000 cells. Together, these results map the context–accuracy trade-offs of single-cell tokenization and show that preserving population-level variance through distribution-based tokens offers an efficient, scalable path for multi-cellular transcriptomics models.

A natural next step is to use MDE as the tokenizer for a transcriptomic foundation model pretrained on diverse cohorts at fixed sequence length M. Shared gene embeddings let pretraining mix co-expression modules, curated gene sets, and individual genes as named channels, with self-supervision on distribution tokens (e.g., identity-conditioned masked prediction) rather than per-cell gene sequences. Relative to gene-level foundation models, this would enable multi-cellular applications—disease diagnosis, clinical trial outcome prediction, and population-scale perturbation prediction—at cohort sizes where $N{\times}G$ context is prohibitive. We leave transfer under domain shift and matched-compute comparisons to gene-level FMs to future work.

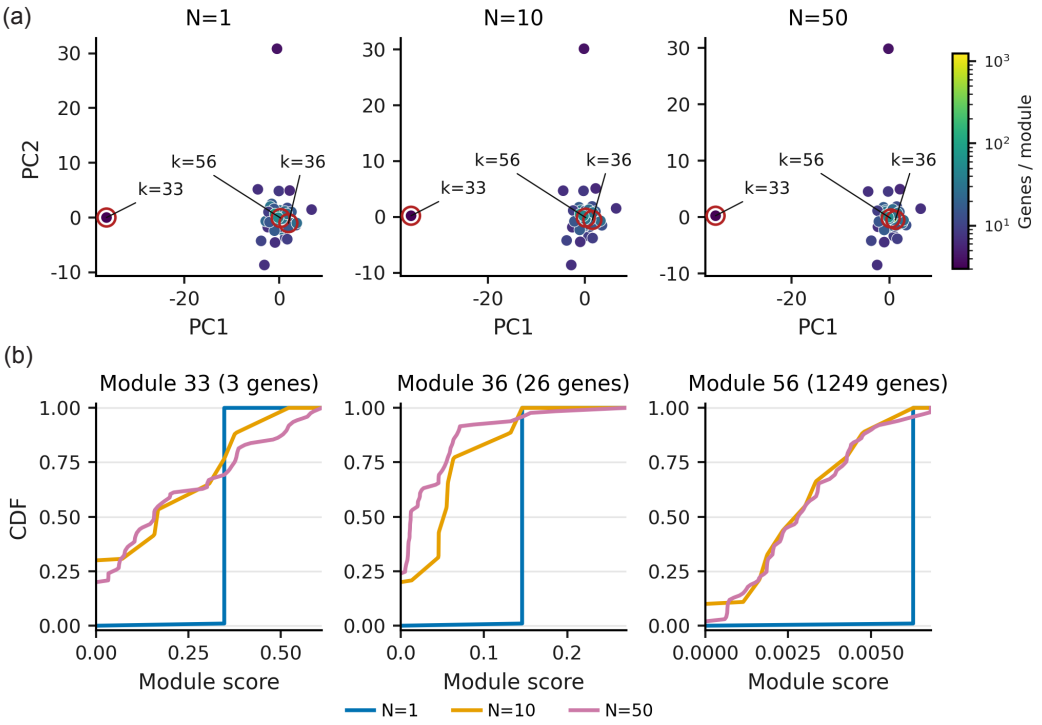


Figure 5: Module identity and distribution tokens for disease prediction (zero-adjusted quantiles, M=100). **(a)** Joint PCA of identity embeddings e_k (mean-pooled gene embeddings) from models trained at $N \in \{1, 10, 50\}$, projected into a shared 2D basis; point color denotes genes per module (log scale). Highlighted modules $k \in \{33, 36, 56\}$ span small to large membership. Embeddings are nearly identical across N, indicating that identity geometry is largely stable to training cohort size and depends on gene membership. **(b)** Zero-adjusted quantile CDFs of module scores for the same three modules in one training sample as N increases. Unlike identity, the distribution cue changes with N: N=1 is a step at a single score, while larger N yield smoother empirical CDFs. A stable identity cue and an N-dependent distribution cue supply complementary information that is summed into each module token $t_k = e_k + \mathrm{Linear}(d_k)$.

AI USE STATEMENT

In this work, we used generative AI tools for assistance in editing software code for implementation of methods within the paper as well as draft and edit parts of the paper. We have not used generative AI tools for design of the method or any conceptual frameworks, and the rest of the required disclosure tasks are not applicable to this work.

We have reviewed all AI-assisted work by reviewing and editing paper sections and software code edited by AI. AI-generated code was verified and tested for correctness by multiple authors, and references were manually sourced by the paper authors, not by AI. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of AI.

REPRODUCIBILITY STATEMENT

All experiments and software code generated for this project is organized in GitHub, and will be fully released for experiment reproducibility after acceptance. We are fully committed to experiment reproducibility, including reporting all metrics for experiments and providing adequate instructions for running the experiments as described in the paper.

REFERENCES

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

Michael Ashburner, Catherine A Ball, Judith A Blake, David Botstein, Heather Butler, J Michael Cherry, Allan P Davis, Kara Dolinski, Selina S Dwight, Janan T Eppig, et al. Gene ontology: tool for the unification of biology. *Nature genetics*, 25(1):25–29, 2000.

Sebastian Birk, Irene Bonafonte-Pardàs, Adib Miraki Feriz, Adam Boxall, Eneritz Agirre, Fani Memi, Anna Maguza, Anamika Yadav, Erick Armingol, Rong Fan, et al. Quantitative characterization of cell niches in spatially resolved omics data. *Nature genetics*, 57(4):897–909, 2025.

Helena L Crowell, Charlotte Soneson, Pierre-Luc Germain, Daniela Calini, Ludovic Collin, Catarina Raposo, Dheeraj Malhotra, and Mark D Robinson. Muscat detects subpopulation-specific state transitions from multi-sample multi-condition single-cell transcriptomics data. *Nature communications*, 11(1):6077, 2020.

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*, pp. 1–11, 2024.

Xiaofeng Dai and Li Shen. Advances and trends in omics technology development. *Frontiers in medicine*, 9:911861, 2022.

César Domínguez Conde, Chao Xu, Louie B Jarvis, Daniel B Rainbow, Sara B Wells, Tamir Gomes, Sarah K Howlett, Ondrej Suchanek, Krzysztof Polanski, Hamish W King, et al. Cross-tissue immune cell analysis reveals tissue-specific features in humans. *Science*, 376(6594):eabl5197, 2022.

Mingze Dong, Abhinav Adduri, Dhruv Gautam, Christopher Carpenter, Rohan Shah, Chiara Ricci-Tam, Yuval Kluger, Dave P Burke, and Yusuf H Roohani. Stack: In-context learning of single-cell biology. *bioRxiv*, 2026.

Gabriela Rapozo Guimarães, Giovanna Resk Maklouf, Cristiane Esteves Teixeira, Leandro de Oliveira Santos, Nayara Gusmão Tessarollo, Nayara Evelin de Toledo, Alessandra Freitas Serain, Cristóvão Antunes de Lanna, Marco Antônio Pretti, Jéssica Gonçalves Vieira da Cruz, et al. Single-cell resolution characterization of myeloid-derived cell states with implication in cancer outcome. *Nature Communications*, 15(1):5694, 2024.

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

Bryan He, Matthew Thomson, Meena Subramaniam, Richard Perez, Chun Jimmie Ye, and James Zou. Cloudpred: Predicting patient phenotypes from single-cell rna-seq. In *PACIFIC SYMPOSIUM ON BIOCOMPUTING 2022*, pp. 337–348. World Scientific, 2021a.

Shanshan He, Ruchir Bhatt, Brian Birditt, Carl Brown, Emily Brown, Kan Chantranuvatana, Patrick Danaher, Dwayne Dunaway, Brian Filanoski, Ryan G Garrison, et al. High-plex multiomic analysis in ffpe tissue at single-cellular and subcellular resolution by spatial molecular imaging. *BioRxiv*, pp. 2021–11, 2021b.

Longda Jiang, Zhixin Cyrillus Tan, Isabella N Grabski, Yuhan Hao, Nathan Nakatsuka, Sourav Sarkar, Anagha Shenoy, and Rahul Satija. Reconstructing developmental and disease progression with sample-level embeddings. *bioRxiv*, pp. 2025–12, 2025.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Dominik Klein, Jonas Simon Fleck, Daniil Bobrovskiy, Lea Zimmermann, Sören Becker, Alessandro Palma, Leander Dony, Alejandro Tejada-Lapuerta, Guillaume Huguet, Hsiu-Chuan Lin, et al. Cellflow enables generative single-cell phenotype modeling with flow matching. *bioRxiv*, pp. 2025–04, 2025.

Daniel Levine, Syed Asad Rizvi, Sacha Lévy, Nazreen Pallikkavaliyaveetil, David Zhang, Xingyu Chen, Sina Ghadermarzi, Ruiming Wu, Zihe Zheng, Ivan Vrkic, et al. Cell2sentence: Teaching large language models the language of biology. *bioRxiv*, pp. 2023–09, 2023.

Tianyu Liu, Edward De Brouwer, Archit Verma, Alsu Missarova, Tony Kuo, Nathaniel Diamant, Hanchen Wang, Minsheng Hao, Tommaso Biancalani, Hector Corrada Bravo, et al. Learning multi-cellular representations of single-cell transcriptomics data enables characterization of patient-level disease states. *Cell Systems*, 17(5), 2026.

Yuzhen Mao, Yen-Yi Lin, Nelson KY Wong, Stanislav Volik, Funda Sar, Colin Collins, and Martin Ester. Phenotype prediction from single-cell rna-seq data using attention-based neural networks. *Bioinformatics*, 40(2):btae067, 2024.

CZI Cell Science Program, Shibla Abdulla, Brian Aevermann, Pedro Assis, Seve Badajoz, Sidney M Bell, Emanuele Bezzi, Batuhan Cakir, Jim Chaffer, Signe Chambers, et al. Cz cellxgene discover: A single-cell data platform for scalable exploration, analysis and modeling of aggregated data. *Nucleic Acids Research*, pp. gkae1142, 2024.

Jingyang Qian, Xin Shao, Hudong Bao, Yin Fang, Wenbo Guo, Chengyu Li, Anyao Li, Hua Hua, and Xiaohui Fan. Identification and characterization of cell niches in tissue from spatial omics data at single-cell resolution. *Nature Communications*, 16(1):1693, 2025.

Aviv Regev, Sarah A Teichmann, Eric S Lander, Ido Amit, Christophe Benoist, Ewan Birney, Bernd Bodenmiller, Peter Campbell, Piero Carninci, Menna Clatworthy, et al. The human cell atlas. *elife*, 6:e27041, 2017.

Syed Asad Rizvi, Daniel Levine, Aakash Patel, Shiyang Zhang, Eric Wang, Curtis Jamison Perry, Ivan Vrkic, Nicole Mayerli Constante, Zirui Fu, Sizhuang He, et al. Scaling large language models for next-generation single-cell analysis. *BioRxiv*, pp. 2025–04, 2026.

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

Jordan W Squair, Matthieu Gautier, Claudia Kathe, Mark A Anderson, Nicholas D James, Thomas H Hutson, Rémi Hudelle, Taha Qaiser, Kaya JE Matson, Quentin Barraud, et al. Confronting false discoveries in single-cell differential expression. *Nature communications*, 12(1):5692, 2021.

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

Fan Yang, Wenchuan Wang, Fang Wang, Yuan Fang, Duyu Tang, Junzhou Huang, Hui Lu, and Jianhua Yao. scbert as a large-scale pretrained deep language model for cell type annotation of single-cell rna-seq data. *Nature machine intelligence*, 4(10):852–866, 2022.

A MULTI-CELLULAR TASK SETUP

The following section details methods for each individual multi-cellular prediction task.

A.1 DISEASE PREDICTION

The disease prediction task is constructed from the Guimaraes et al. myeloid-derived single-cell transcriptomics dataset (Guimarães et al., 2024), comprising of 215,337 cells across 51 donors with 15 annotated disease conditions. We restrict the dataset to cells sequenced with the 10x 3' v2 assay, leaving out data sequenced by other assays in order to reduce batch effects among sequencing methods. We removed cells with fewer than 200 expressed genes, and removed genes expressed in fewer than 100 cells, leaving us with 215,108 cells and 21,843 genes. We performed library-size normalization to 10,000 counts followed by log1p transformation. We then identify 5,000 highly variable genes (HVGs) for tokenization methods which require HVGs to be calculated. Because many of the 15 diseases in the dataset are only present in 1 patient, a patient-level split was not possible; we instead perform a cell-level split in order to retain all 15 disease categories. For each donor-disease pair, cells are divided into an 80% training partition and a 20% test partition through stratified random sampling. This partition structure ensures that all 51 donors and all 15 disease labels remain represented in both training and test cell pools.

Multi-cell groupings for samples are formed by sampling N cells with replacement from a single donor with a specific disease label. For each eligible donor-disease pair, 1,000 training groups are drawn from the patient's training cells, and 1,000 test groups are drawn from the test cells. Each draw samples a maximum cohort size of 1,000 cells. Smaller cohort sizes N in 1, 3, 5, 10, 20, 50, 100, 200, 500, 1000 are obtained by taking subsets of the larger group. A validation set of 1,000 groups is held out from the generated training groups, with the constraint that every disease class contains at least one representative sample.

The prediction objective is formulated as multi-class classification across the 15 global disease classes. Given an input sequence representing N cells from a given tokenization framework, the model extracts the final-layer CLS token embedding and passes it to a linear classification head. This head outputs logits for all 15 target categories. Training optimizes standard categorical cross-entropy loss against the ground-truth disease label. Evaluation metrics are calculated only on the test set, including: classification accuracy, balanced accuracy, macro-F1, micro-F1, weighted-F1, and cross-entropy.

A.2 CELL COMPOSITION PREDICTION

The cell composition prediction benchmarking task is constructed on the cross-tissue immune dataset from Dominguez et al. (Domínguez Conde et al., 2022), which contains 329,762 cells taken from 12 distinct donors with 35 unique annotated immune cell subtypes. We perform standard preprocess on the raw dataset, filtering out cells with fewer than 200 expressed genes and genes expressed in fewer than 10 cells. This leaves us with 329,762 cells and 27,958 genes. Raw counts are library-size normalized to 10,000 counts per cell, and undergo a log transformation. To maintain well-defined probability targets across all subjects, the cell-type space is restricted to the 28 subtypes shared across all 12 donors, so that a donor-level split does not leave certain cell types only in a few donors. A donor-level split randomly splits donors into a training group of 8 donors (171,441 cells), 2 validation donors (34,737 cells), and 2 held-out test donors (123,584 cells), preventing data leakage across patient cells.

To benchmark models across cohort sizes $N \in \{20, 50, 100, 200, 500, 1000\}$, we draw 10,000 multi-cell samples per each donor, yielding 80,000 training, 20,000 validation, and 20,000 test samples per each value of N. For a given cell type proportion prediction sample, we first sample a target cell type proportion vector, which we try to match by sampling multi-cell groups from the data from that patient. This target cell type proportion vector is drawn from a symmetric Dirichlet prior ($\alpha = 0.4$) blended with a random dominant subtype weight (30% to 60%), resulting in a cell type proportion distribution that is dominant in 1-2 cell types, and contains trace amounts of a few other cell types, forcing models to learn cell type composition that varies in cell type dominance, rather than randomly sampling which would simply approximate the true cell type distribution with little variance. Sparsity artifacts can potentially be introduced at larger cell counts (N), since trace amount

of cells from rare cell types would appear in large multi-cell groups (N=1000), resulting in a less-sparse target probability vector compared to smaller values of N. To prevent these sparsity artifacts in rare cell types, we introduce a sparsity-locking procedure: cell types that have zero proportion at the $N = 20$ cell draw are masked from sampling at larger values of N, therefore maintaining the sparsity level across different cohort sizes (N) and keeping the task relatively similar in target vector sparsity among values of N. For every scale of N, we sample cells to best approximate the target probability vector and best match the N=20 coarsest cell type proportion draw.

The training objective for cell type composition requires optimizing models to predict the empirical ground-truth probability distribution over the 28 shared cell types in the multi-cell sample. The model pools contextual cell or module tokens via a learned CLS token, and a linear head outputs an unnormalized 28-dimensional logit vector. Models are trained by minimizing the soft cross-entropy loss against the true cell-type distribution. Performance is evaluated across multiple distribution and ranking metrics, including: soft cross-entropy, root mean squared error (RMSE), Jensen-Shannon divergence, Kullback-Leibler (KL) divergence, top-k classification accuracy ($k \in \{1, 3, 5\}$), and mean per-sample Spearman rank correlation across all classes and top-k subsets ($k \in \{5, 10, 15\}$). We show top-15 spearman correlation and JS-divergence as representative metrics in our main results.

A.3 Niche Prediction

The spatial niche prediction benchmark is constructed on the CosMx Spatial Molecular Imager (SMI) human liver dataset (He et al., 2021b), which profiles single-cell spatial transcriptomics across a panel of 1,000 curated genes in a healthy and a diseased liver tissue donor. We restrict our benchmark to the healthy liver tissue comprising 340,517 cells. Quality control filters exclude cells with fewer than 50 detected genes and remove genes detected in fewer than 3 cells. The target prediction space corresponds to the 6 annotated functional microenvironment niches of the healthy liver: Central Vein, Portal, Zone 1, Zone 2a, Zone 2b, and Zone 3a. To prevent data leakage between neighboring spatial contexts, the dataset was split into training and test sets based on spatial coordinates.

Multi-cellular cohorts of varying scale $N \in \{1, 5, 10, 20, 50, 100, 200, 500\}$ are generated based on local spatial proximity to capture multi-cellular context. For each target sample, an anchor cell is selected, and its $N-1$ spatial nearest neighbors are retrieved using a k-d tree constructed on the two-dimensional spatial coordinates (x, y). Similar to the prefix-based design in disease classification, multi-cell cohorts across varying N are constructed by indexing the nearest spatial neighbors in ascending order of Euclidean distance from the anchor cell, guaranteeing that smaller cohort sizes represent strict spatial subsets of the larger neighborhoods. The target label is formulated as the empirical niche composition of the cohort: for a group of N cells, the ground truth target is a 6-dimensional probability simplex representing the relative proportion of each spatial niche label across the sampled cells.

The prediction objective mirrors the formulation used in cell composition estimation. Given an input sequence encoding the N neighboring cells or their aggregated module features, the model extracts the final-layer CLS token embedding and feeds it into a linear projection head. This head outputs an unnormalized 6-dimensional logit vector corresponding to the annotated spatial niches. The network is trained by minimizing the soft cross-entropy loss between the predicted softmax distribution and the empirical ground-truth niche composition. Model performance across cohort scales is evaluated using soft cross-entropy, root mean squared error (RMSE), Jensen-Shannon divergence, Kullback-Leibler (KL) divergence, top-k classification accuracy ($k \in \{1, 3\}$), and mean per-sample Spearman rank correlation across niche proportions.

B Training Hyperparameters

B.1 Disease Prediction

Module Distribution Encoding (MDE). MDE models summarize the expression profiles of cohort cells into $M = 100$ co-expression modules, each encoded as a $d = 100$ dimensional distribution token ($K = 100$ for zero-adjusted percentile encodings). The transformer backbone uses a 4-layer

architecture with hidden dimension $d_{\text{model}} = 128$, 8 attention heads, a feedforward expansion dimension of $d_{\text{ff}} = 256$, and dropout of 0.1. The maximum sequence length is set to accommodate the K module tokens (or up to $22,000$ for full-feature zero-adjusted CDF variants). Training uses the AdamW optimizer with a learning rate of 1×10^{-4}, weight decay of 0.01, and a linear warmup schedule with a warmup ratio of 0.05 across 20 epochs. Overall batch size after gradient accumulation for all models is set to 64, and training is done using bfloat16 mixed precision. Checkpoint selection is done by minimum validation cross-entropy, evaluated every 200 steps.

Baseline Models. For the cell projection baseline, each cell expression vector is projected via a single linear layer into a continuous hidden token embedding. The 4-layer transformer backbone keeps $d_{\text{model}} = 128$, 8 attention heads, $d_{\text{ff}} = 256$, and dropout of 0.1, with sequence length matching cohort size N ($= N$). Models are trained for 40 epochs with a learning rate of 1×10^{-4}, a per-device batch size of 64, and gradient accumulation of 1. For foundation model baselines with gene-level tokenization (scGPT and Geneformer), each cell is allocated $1,000$ gene tokens separated by boundary tokens, scaling sequence context to $N \times (1000 + 1)$. scGPT employs 51 discrete expression bins with random gene selection, while Geneformer applies target sum library normalization $(10,000)$ followed by gene ranking. These architectures use $d_{\text{model}} = 128$, 8 attention heads, $d_{\text{ff}} = 256$, and dropout of 0.05. Because of extreme context lengths, scGPT and Geneformer are trained with a per-device batch size of 4 and 16 gradient accumulation steps (effective batch size of 64) over 15 epochs using bfloat16 precision.

B.2 Cell Composition Prediction

Module Distribution Encoding (MDE). MDE configurations evaluate both zero-adjusted percentile distribution tokens ($K = 50$ percentiles per token). For transformer-based MDE models, the standard 4-layer transformer architecture operates with $d_{\text{model}} = 128$, 8 heads, $d_{\text{ff}} = 256$, and dropout of 0.05, using maximum sequence lengths up to 1,024 (and up to 20,000 for full-gene variants). A linear projection maps continuous feature vectors to d_{model}. For MLP baselines trained on pseudobulk or flattened MDE vectors, the architecture uses hidden dimensions of $[512, 256]$ with dropout of 0.05. All models are trained for 20 epochs using AdamW with an initial learning rate of 1×10^{-4}, weight decay of 0.01, and a linear warmup ratio of 0.05. The effective batch size is 64. Models are evaluated every 50 steps, with early stopping and checkpoint selection through minimum validation cross entropy loss.

Baseline Models. The cell projection baseline maps each cell directly to $d_{\text{model}} = 128$, maintaining context length at N. It adopts identical backbone transformer parameters ($L = 4, H = 8, d_{\text{ff}} = 256$, dropout 0.05). For scGPT and Geneformer, each cell is tokenized using up to 1,000 gene tokens, scaling total sequence length to $N \times 1001$. scGPT applies 51 expression bins and random gene sampling, whereas Geneformer applies non-parametric rank-value encoding. Both baselines use identical backbone dimensions ($d_{\text{model}} = 128, H = 8, d_{\text{ff}} = 256$, dropout 0.05). Because of large context sizes at $N \geq 20$, training executes with per-device batch sizes of 4 and 16 gradient accumulation steps (effective batch size 64) under bfloat16 precision across 20 epochs. Models are evaluated every 50 steps, with early stopping and checkpoint selection.

B.3 Niche Prediction

Module Distribution Encoding (MDE). MDE models for spatial niche prediction are constructed with $M = 50$ co-expression gene modules, each represented by a $d = 100$ dimensional distribution token derived from zero-adjusted quantile encodings. The backbone transformer employs 4 layers, a model dimension $d_{\text{model}} = 256$, 8 attention heads, a feedforward dimension $d_{\text{ff}} = 512$ (or default $4 \times d_{\text{model}}$), and transformer dropout of 0.1. Positional embeddings are explicitly disabled, treating module tokens as an unordered set. Training optimizes soft cross-entropy across the 6 niche proportion targets over 20 epochs using the AdamW optimizer. Training uses a learning rate of 5×10^{-5}, weight decay of 0.01, warmup ratio of 0.05, a per-device batch size of 64 with gradient accumulation of 1 (effective batch size of 64), and FP16 mixed precision. Model evaluation and checkpointing are conducted every 200 steps.

Baseline Models. All baseline models share a 4-layer transformer backbone with $d_{\text{model}} = 256$, 8 heads, $d_{\text{ff}} = 512$, and dropout of 0.1. For cell projection, the input sequence length is capped at the neighborhood cell count N. Training runs with a per-device batch size of 64 and gradient

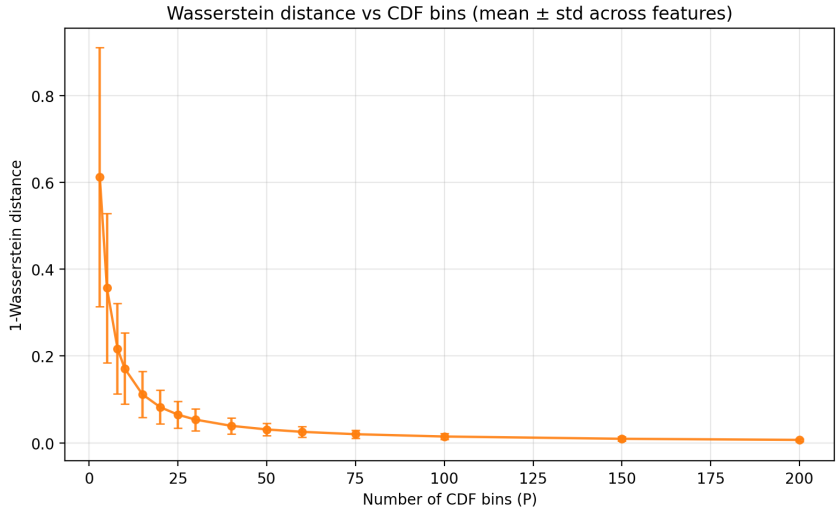


Figure 6: Elbow plot for the optimal number of quantile bins to use for the Guimaraes disease prediction dataset. The number of quantile bins (x-axis) versus the divergence of real data versus data sampled from the fitted CDF distribution is shown. The elbow of the plot is located roughly at 25 bins, however we use 50 bins in our experiments to capture more information since it does not increase token usage.

accumulation of 1. For single-cell tokenization baselines (scGPT and Geneformer), each cell in the local neighborhood is budgeted up to 300 gene tokens plus separator tokens, with context length bounded by $N \times (300 + 2) + 8$. scGPT utilizes random gene selection across the 1,000 measured CosMx genes. Training for scGPT and Geneformer employs a learning rate of 5×10^{-5}, weight decay of 0.01, a per-device batch size of 32, and 2 gradient accumulation steps (effective batch size of 64) with FP16 precision over 20 epochs. Checkpoints are evaluated and saved every 500 steps, selected by validation loss.

C Full Result Tables

Full metric tables for all main experiments are presented in Tables 1, 2, and 3.

Tokenization	Metric	Number of Cells (N)					
		1	3	5	10	20	50
Cell Projection	F1 (↑)	0.859 ± 0.010	0.951 ± 0.004	0.961 ± 0.001	0.968 ± 0.001	0.968 ± 0.001	0.971 ± 0.002
	Acc (↑)	0.738 ± 0.032	0.906 ± 0.010	0.924 ± 0.004	0.933 ± 0.000	0.933 ± 0.000	0.933 ± 0.001
scGPT	F1 (↑)	0.842 ± 0.006	0.930 ± 0.003	0.955 ± 0.003	0.969 ± 0.008	0.972 ± 0.005	–
	Acc (↑)	0.726 ± 0.007	0.858 ± 0.007	0.903 ± 0.009	0.934 ± 0.019	0.937 ± 0.011	–
Geneformer	F1 (↑)	0.840 ± 0.002	0.931 ± 0.000	0.947 ± 0.002	0.968 ± 0.005	0.965 ± 0.004	–
	Acc (↑)	0.732 ± 0.002	0.864 ± 0.002	0.896 ± 0.003	0.929 ± 0.008	0.935 ± 0.011	–
MDE	F1 (↑)	0.806 ± 0.005	0.927 ± 0.003	0.960 ± 0.001	0.978 ± 0.002	$\mathbf{0.987 \pm 0.004}$	0.985 ± 0.002
	Acc (↑)	0.691 ± 0.010	0.867 ± 0.002	0.927 ± 0.002	0.972 ± 0.002	0.990 ± 0.000	$\mathbf{0.996 \pm 0.000}$

Table 1: Weighted F1 and balanced accuracy across varying numbers of cells for disease prediction. Values indicate mean ± standard deviation across seeds.

Tokenization	Metric	Number of Cells (N)							
		1	5	10	20	50	100	200	500
Cell Projection	Spearman (↑)	0.476	0.619	0.674	0.736	0.804	0.845	0.870	**0.899**
	JS Divergence (↓)	0.267	0.166	0.152	0.117	0.076	0.053	0.035	0.019
	Top-1 Acc (↑)	0.475	0.546	0.578	0.601	0.627	0.622	0.583	0.590
Geneformer	Spearman (↑)	0.465	0.603	0.675	0.745	0.819	OOM	OOM	OOM
	JS Divergence (↓)	0.278	0.181	0.140	0.101	0.059	OOM	OOM	OOM
	Top-1 Acc (↑)	0.464	0.520	0.584	0.641	0.708	OOM	OOM	OOM
scGPT	Spearman (↑)	0.471	0.618	0.684	0.750	0.825	OOM	OOM	OOM
	JS Divergence (↓)	0.269	0.169	0.132	0.095	0.054	OOM	OOM	OOM
	Top-1 Acc (↑)	0.465	0.551	0.602	0.659	**0.734**	OOM	OOM	OOM
MDE	Spearman (↑)	0.422	–	0.665	–	0.817	0.857	0.882	0.896
	JS Divergence (↓)	0.307	–	0.149	–	0.061	0.040	0.027	**0.018**
	Top-1 Acc (↑)	0.403	–	0.550	–	0.685	0.695	0.676	0.602

Table 2: Spearman correlation, Jensen-Shannon divergence, and Top-1 accuracy across varying numbers of cells for niche prediction.

Tokenization	Metric	Number of Cells (N)					
		20	50	100	200	500	1000
Cell Projection	Top-15 Spearman (↑)	0.788 ± 0.012	0.852 ± 0.005	0.865 ± 0.011	0.888 ± 0.003	$\mathbf{0.897 \pm 0.002}$	0.894 ± 0.004
	JS Divergence (↓)	0.082 ± 0.012	0.050 ± 0.000	0.045 ± 0.009	0.033 ± 0.003	$\mathbf{0.028 \pm 0.000}$	0.029 ± 0.002
scGPT	Top-15 Spearman (↑)	0.521 ± 0.010	OOM	OOM	OOM	OOM	OOM
	JS Divergence (↓)	0.169 ± 0.001	OOM	OOM	OOM	OOM	OOM
Geneformer	Top-15 Spearman (↑)	0.607 ± 0.017	OOM	OOM	OOM	OOM	OOM
	JS Divergence (↓)	0.143 ± 0.002	OOM	OOM	OOM	OOM	OOM
MDE	Top-15 Spearman (↑)	0.602	0.653	0.684	0.705	0.734	0.747
	JS Divergence (↓)	0.131	0.106	0.096	0.091	0.083	0.079

Table 3: Top-15 Spearman correlation and Jensen-Shannon divergence across varying numbers of cells for cell composition. Values indicate mean across seeds ± standard deviation.