PROTOTYPE GUIDED POST-PRETRAINING FOR SINGLE-CELL REPRESENTATION LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Single-cell representation learning (SCRL) from gene expression data offers a way to uncover the complex regulatory logic underlying cellular function. Inspired by large language models in natural language modeling, several single-cell pretrained models have recently been proposed that treat genes as tokens and cells as sentences. However, these models are fundamentally limited by the long-tailed nature of cell-type distributions and struggle to generalize under covariate shifts in gene expression data. While fine-tuning is often used to mitigate these issues, we observe that performance remains bounded. To address this challenge, we introduce *CellRefine*, a post-pretraining method that operates between the pretraining and fine-tuning stages of a single-cell foundation model. *CellRefine* uses a multi-faceted objective that incorporates marker-gene sets as structural priors to guide post-pretraining and refine the latent embedding manifold of cells. Across multiple computational biology tasks, empirical results show that *CellRefine* consistently improves downstream performance, yielding gains up to 15%.

1 INTRODUCTION

Advances in machine learning are rapidly transforming our ability to study complex cellular systems, providing powerful computational frameworks for decoding high-dimensional single-cell data and enabling breakthroughs in precision diagnostics Xia et al. (2025), personalized therapies Tan et al. (2025), and proactive disease prevention Shmatko et al. (2025). Among the most promising directions is single-cell representation learning (SCRL), which leverages cell–gene expression data derived from modalities such as single-cell RNA sequencing and spatial transcriptomics to learn rich representations of cell identities and functions.

Inspired by the success of large language models in natural language modeling, recent work has proposed single-cell pretrained models for SCRL, framing genes as tokens and cells as sentences Theodoris et al. (2023); Cui et al. (2024). These models open new opportunities for uncovering latent cellular states and disentangling complex gene regulatory programs, with applications ranging from reconstructing developmental trajectories and revealing disease mechanisms to predicting drug-induced perturbations for therapeutic discovery.

Despite this widespread applicability, existing pretrained models for SCRL struggle to learn equally rich representations across the full spectrum of cell types Navidi et al. (2025) and are generally performance-bound. We hypothesize that these limitations stem from two primary factors. First, the long-tailed distributions inherent in single-cell datasets lead to noisy or collapsed embeddings for underrepresented cell populations. For example, Figure 1 illustrates the extreme disparity in sample counts between cell types in a standard blood cell dataset Zheng et al. (2017), with further analyses provided in Appendix J. Second, pervasive covariate shifts in gene expression measurements create significant distribution gaps between training and inference data. These technical changes often mask

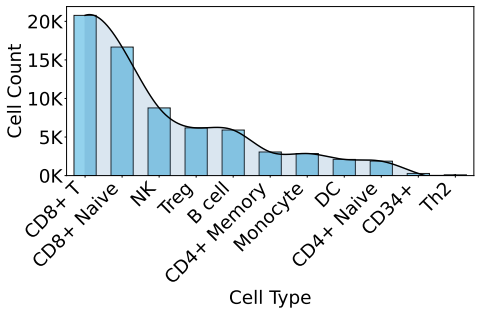


Figure 1: Long-tail distribution of cell types in the blood cell dataset Zheng et al. (2017).

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.

true biological signals, forcing models to overfit batch-specific patterns, challenging robust downstream generalization.

The extreme imbalance between cell types reflects labeling challenges, technical artifacts, and rare biology, with critical but rare populations, such as disease-initiating stem cells or specialized immune cells, often constitute less than 0.1% of a sample Azizi et al. (2018); Grün et al. (2015); Villani et al. (2017). Our analysis confirms this long-tail pattern across major public benchmarks, revealing power-law tail exponents as low as $\alpha \approx 0.2-0.6$ (see Appendix I). Such a severe skew biases pretrained models toward abundant cell types, causing the latent space to overlook rare but clinically significant cell populations.

Even when underrepresented cells are captured, their biological signal is frequently buried under pervasive covariate shifts. Driven by technical artifacts like batch effects and varying sequencing depths, these shifts alter the marginal distribution of gene expression levels without changing the underlying cell-type identity. Because most single-cell pretrained models are sensitive to these high-dimensional fluctuations, they often fail to generalize across different sequencing platforms or experimental protocols. Although fine-tuning on individual datasets is commonly used to address these issues, its effectiveness is often limited because it begins from an unstable latent embedding manifold in the foundation model, restricting its ability to learn more refined cell representations.

To address these limitations, we introduce *CellRefine*, a post-pretraining framework for single-cell foundation models that is designed to better capture rare cell types in the long tail of the cell-type distribution while improving robustness to covariate shift. Rather than relying on fine-tuning alone from an unstable latent embedding manifold, *CellRefine* injects biologically grounded inductive biases during post-pretraining to reshape the latent embedding space into a more stable and discriminative structure. Specifically, *CellRefine* leverages mechanistic biological priors derived from known marker gene programs to guide the model toward distinguishing rare cell types whose signals are often overlooked or confounded by technical noise. In addition, to separate closely related cell types within the same parent lineage in the cell ontology, *CellRefine* incorporates a lineage-aware regularization loss that explicitly encourages the separation of cells sharing the same parent lineage.

Together, the marker gene program guidance and cell ontology-aware regularization provide stronger biological grounding and help the model learn more discriminative, stable, and generalizable representations, thereby mitigating both long-tail and covariate-shift challenges. By explicitly incorporating prior knowledge about cell identity and lineage structure, these objectives encourage the learned representation space to better reflect biologically meaningful relationships rather than relying solely on statistical regularities in the pretraining data. More broadly, this work asks the following question: **To what extent can biologically informed inductive biases and explicit learning guidance improve the downstream performance, robustness, and transferability of single-cell foundation models?**

To assess the utility of *CellRefine*, we conduct extensive experiments across three core tasks in computational biology: 1) cell identity prediction 2) spatial transcriptomic imputation and 3) gene perturbation response prediction across 11 datasets spanning blood, pancreas, myeloid, liver, brain, lung, and heart tissues. Empirical evaluation demonstrates that *CellRefine* consistently surpasses state-of-the-art single-cell pretrained models across all three tasks and organ types, yielding up to 15% performance improvements.

2 METHODOLOGY

In this section, we elaborate on the details of our method behind *CellRefine*. Additionally, an overview of the proposed method as illustrated in Figure 2.

2.1 POST-PRETAINING WITH *CellRefine*

CellRefine is an intermediate post-pretraining technique designed to better align a single-cell pretrained model embedding manifold and prepare it for downstream fine-tuning. Specifically, *CellRefine* is applied after pretraining and before downstream fine-tuning. Such intermediate adaptation strategies have recently been proposed in computer vision as an effective way to refine pretrained models for specific data characteristics and downstream objectives Yamaguchi et al. (2025).

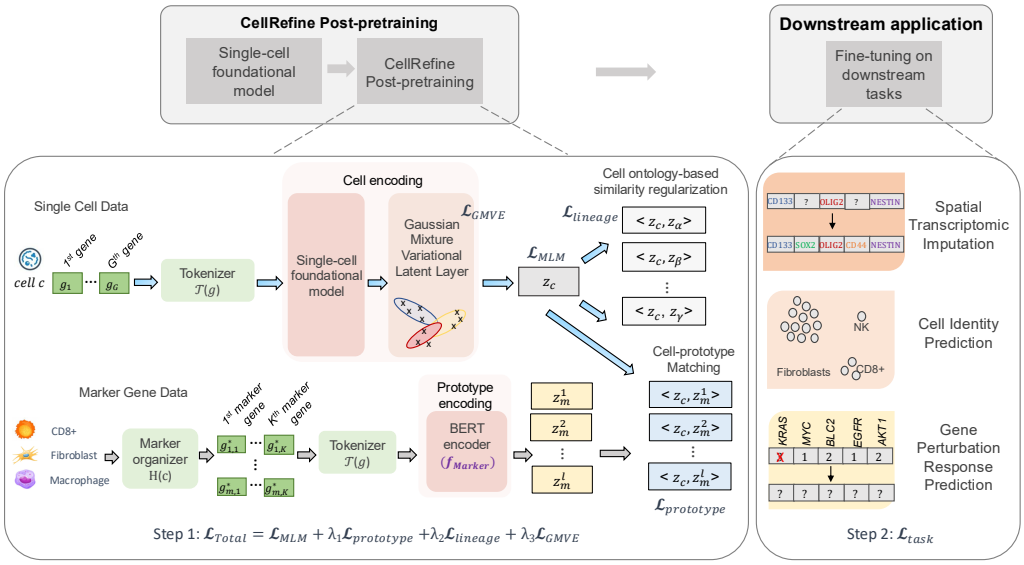


Figure 2: Overview of *CellRefine*, a post-pretraining method for single-cell foundation models. *CellRefine* refines the latent cell embedding space of a given foundation model on a target cell dataset before task-specific fine-tuning.

Specifically, let $f_{\theta^{\text{pt}}}$ denote a single-cell pretrained model after the initial pretraining stage, where pretrained θ^{pt} are often obtained by optimizing a language modeling objective (refer to Appendix C for the details of pretraining objective formulation). We then apply *CellRefine* as an intermediate post-pretraining stage to adapt this pretrained model towards a given dataset, yielding an updated model $f_{\theta^{\text{pp}}}$, where θ^{pp} is initialized from θ^{pt} and optimized under our post-pretraining method. Finally, for a downstream task of interest, we fine-tune the post-pretrained model to obtain $f_{\theta^{\text{ft}}}$, where θ^{ft} is initialized from θ^{pp} and optimized using the task-specific objective. Thus, the overall training setup can be written as,

$$f_{\theta^{\text{pt}}} \xrightarrow{\textit{CellRefine}} f_{\theta^{\text{pp}}} \xrightarrow{\text{Downstream Fine-tuning}} f_{\theta^{\text{ft}}}$$

2.2 Prototype-guided Learning

The *CellRefine* post-pretraining stage is driven by our novel *prototype-guided Learning* mechanism, a training method that grounds a given pretrained model on mechanistic biological priors to guide the learning towards refined cell representations overall. To this end, we introduce two biological priors to the learning process: 1) gene marker program-based guided masked language modeling and 2) cell similarity regularization based on cell ontology.

2.2.1 Gene marker program-based guided masked language modeling

Gene markers are specific genes whose expression patterns are characteristic of particular cell types, effectively serving as biological "labels" that distinguish one cell from another. In single-cell analysis, researchers often rely on these well-established markers to manually identify and classify cells, interpreting clusters or assigning cell types based on known marker genes Dumitrascu et al. (2021). This suggests that incorporating marker genes during the post-pretraining stage can provide a biologically meaningful supervisory signal, helping to further structure the pretrained embedding space.

However, it is important to note that the full set of marker genes for a given cell type is typically unknown; only a subset of markers is often identified. Consequently, marker genes provide a partial learning signal rather than a complete one. Nevertheless, known markers capture key functional and identity-related features of cells, making them a useful source of guidance. In *CellRefine*, we tap on this unexplored opportunity to further refine the embeddings of a pretrained model.

To start, we construct marker gene sets for each cell type. For this, we integrate multiple publicly available marker gene resources as described in Appendix B. In short, for each cell type c_i where

$i \in \{1, \ldots, l\}$ where l is the number of cell types in a given dataset, we identify K marker genes and order them based on their position in the cell ontology using a marker organizer function $H(c_i)$ (refer to Appendix B for the implementation details of $H(c_i)$). We refer to this ordered sequence of marker genes as a *prototype* of the cell type. Formally, the output of $H(c_i)$ is an ordered sequence of marker genes $(g^*_{i,1}, \ldots, g^*_{i,K})$ for the cell type c_i.

These ordered marker genes are tokenized using the tokenizer $\tau(\cdot)$ and then encoded with a learnable model $f_{\text{marker}}(\cdot)$. We call this process *prototype encoding*, which produces an embedding for each cell-type prototype and provides a biologically informed reference in the latent space.

$$z^i_m = f_{\text{marker}}(\tau(g^*_{i,1}, \ldots, g^*_{i,K})) \tag{1}$$

We perform this procedure for each cell type in a given dataset to get all prototype embeddings $\{z^1_m, \ldots, z^l_m\}$. Then we define a *prototype-guided regularization loss* $\mathcal{L}_{c \rightleftarrows p}$ to use them for guided post-pretraining. Specifically, let z^i_c denote the embedding of cell c_i obtained from a given single-cell pretrained model. We compute the cosine similarity between the cell embedding z^i_c and each prototype embedding z^j_m.

$$s_{i,j} = \cos(z^i_c, z^j_m) = \frac{z^i_c \cdot z^j_m}{\|z^i_c\| \, \|z^j_m\|} \tag{2}$$

We then define a cross-entropy loss using these similarities, treating the true prototype of the cell as the correct label.

$$\mathcal{L}_{prototype} = -\sum_{i=1}^{m} \log \frac{\exp(s_{i,i})}{\sum_{j=1}^{m} \exp(s_{i,j})} \tag{3}$$

This regularization encourages each cell embedding z^i_c to align closely with its corresponding prototype z^i_m than with other prototypes, injecting biologically meaningful structure into the representation space. As illustrated in Figure 2, the ground-truth cell-type label selects the prototype, while its marker genes define the prototype structure; the prototype loss is therefore a supervised objective.

2.2.2 Cell similarity regularization based on cell ontology

While prototype-guided training helps the model capture inter-cell-type differences, cell types that share the same parent lineage in the cell ontology may still remain insufficiently separated in the embedding space. This is particularly challenging for closely related cell types, whose transcriptional programs often exhibit substantial overlap. To further refine these local structures, ***CellRefine*** incorporates a lineage-aware regularization loss that explicitly encourages the separation of cells that share the same parent lineage and ontology level.

Specifically, let $\mathcal{R}$ denote the set of cell-type pairs (c_i, c_j) such that c_i and c_j share the same parent lineage and belong to the same ontology level. During training, for each batch, we identify all pairs in $\mathcal{R}$ that are present in the batch and compute the cosine similarity between their cell embeddings,

$$r_{i,j} = \cos(z^i_c, z^j_c) = \frac{z^i_c \cdot z^j_c}{\|z^i_c\| \, \|z^j_c\|} \tag{4}$$

Because these cell types are semantically close in the ontology yet represent distinct biological identities, we seek to prevent their embeddings from collapsing together. We therefore define a lineage-aware separation loss that penalizes excessive similarity among such related cell-type pairs.

$$\mathcal{L}_{\text{lineage}} = \frac{1}{|\mathcal{R}_b|} \sum_{(i,j) \in \mathcal{R}_b} \max(0, r_{i,j} - m) \tag{5}$$

where m is a separation margin and $\mathcal{R}_b \subseteq \mathcal{R}$ denotes the subset of valid cell-type pairs present in the current training batch.

2.2.3 Extending the pretrained model with Gaussian Mixture Variational Encoding

Previous work has shown that explicitly modeling the latent embedding space of pretrained single-cell models as a clustered structure can improve cell representations Wen et al. (2023). Motivated

by this, we further augment the pretrained model with a Gaussian Mixture Variational Encoder (GMVE) Dilokthanakul et al. (2016). Given the pretrained model output $\mathbf{h}_c = f_{\theta^{pt}}(\mathbf{c}_o) \in \mathbb{R}^d$, the GMVE defines the variational posterior $q(\mathbf{z}_c \mid \mathbf{h}_c)$,

$$q(\mathbf{z}_c \mid \mathbf{h}_c) = \sum_{l=1}^{L} \pi_l(\mathbf{h}_c)\mathcal{N}(\mathbf{z}_c \mid \boldsymbol{\mu}_l(\mathbf{h}_c), \boldsymbol{\Sigma}_l(\mathbf{h}_c)) \tag{6}$$

where $\pi_l(\mathbf{h}_c)$ is the mixture weight ($\sum \pi_l = 1$), and $\boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l$ are the parameters of the l-th Gaussian. To regularize this structure, we minimize the KL divergence between the variational posterior and a learned Gaussian mixture prior,

$$\mathcal{L}_{\text{GMVE}} = \text{KL}\left(q(\mathbf{z}_c \mid \mathbf{h}_c) \parallel \sum_{l=1}^{L} p_l \mathcal{N}(\mathbf{z}_c \mid \boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l)\right) \tag{7}$$

2.2.4 Overall post-pretraining objective

Finally, the overall objective of ***CellRefine*** combines the regular pretraining loss $\mathcal{L}_{\text{pre}}$ (refer to Appendix C for preliminaries on pretraining loss), the prototype-guided regularization, the lineage-aware regularization, and the GMVE regularization.

$$\mathcal{L}_{\text{Total}} = \mathcal{L}_{pre} + \lambda_1 \mathcal{L}_{prototype} + \lambda_2 \mathcal{L}_{\text{lineage}} + \lambda_3 \mathcal{L}_{\text{GMVE}} \tag{8}$$

where λ_1, λ_2 and λ_3 control the contributions of the three regularization terms.

3 Experimental Setup

3.1 Evaluation Tasks

We evaluate the utility of ***CellRefine*** on three widely used computational biology tasks: 1) cell identity prediction, 2) spatial transcriptomics imputation, and 3) gene perturbation response prediction.

Cell Identity Prediction: Cell identity prediction is the task of assigning each cell a cell-type label from transcriptomic data. We define two versions of this task, on-domain and out-of-domain prediction. In the on-domain setting, both training and evaluation are performed on single-cell RNA sequencing (scRNA-seq) data. In the out-of-domain setting, models trained on scRNA-seq data are evaluated on spatial transcriptomics data to probe zero-shot out-of-distribution generalization.

We benchmark ten human single-cell transcriptomic datasets, including eight scRNA-seq datasets; peripheral blood Zheng et al. (2017); Gayoso et al. (2022), pancreas Chen et al. (2023), liver Lin et al. (2020), myeloid Cheng et al. (2021), multiple sclerosis Schirmer et al. (2019), heart Tucker et al. (2020), and lung Kim et al. (2020), and two spatial transcriptomics datasets of liver 10x Genomics (2025). Detailed dataset descriptions can be found in Appendix F. We report macro F1 scores for on-domain evaluations and recall@k scores for zero-shot, out-of-domain transfer tasks; recall@k denotes the proportion of test samples whose correct label appears among the top-k retrieved candidates.

Spatial Transcriptomic Imputation: Spatial transcriptomic imputation is the task of predicting unmeasured gene expression from the subset of transcripts observed in each cell. We benchmark this task on the two spatial transcriptomics datasets. We randomly mask 15% of genes as prediction targets and use the remaining genes as input. We evaluate performance using Pearson correlation and cosine similarity between predicted and ground-truth.

Gene Perturbation Response Prediction: Gene perturbation response prediction aims to predict changes in a cell's gene expression resulting from a specific perturbation, given the unperturbed gene expression profile as input. This task is key to developing pharmacological interventions. We formulate this as an out-of-distribution task by holding out one cell type from each reference dataset and using the held-out cell type as the query dataset. We tested this task on a peripheral blood dataset Kang et al. (2018). We use Pearson correlation coefficient to measure the performance of this task.

Method	Blood	Myeloid	Liver	MS	Avg. ↑
PCA + LR	0.586 ± 0.003	0.340 ± 0.001	0.301 ± 0.004	0.708 ± 0.004	0.484 ± 0.003
LR on HVG	0.723 ± 0.007	0.354 ± 0.001	0.323 ± 0.002	0.704 ± 0.006	0.526 ± 0.004
scTOP	0.514 ± 0.004	0.251 ± 0.000	0.306 ± 0.001	0.624 ± 0.001	0.424 ± 0.002
scTOP-rep on HVG + LR	0.739 ± 0.006	0.342 ± 0.001	0.340 ± 0.001	0.705 ± 0.003	0.532 ± 0.003
scANVI	0.632 ± 0.004	0.338 ± 0.006	0.321 ± 0.008	0.742 ± 0.002	0.508 ± 0.005
CellTypist	0.686 ± 0.005	0.341 ± 0.002	0.281 ± 0.004	0.660 ± 0.002	0.492 ± 0.003
scBERT	0.704 ± 0.011	0.315 ± 0.010	0.207 ± 0.021	0.612 ± 0.020	0.460 ± 0.016
scBERT + *CellRefine* (Ours)	0.817 ± 0.012	0.382 ± 0.008	0.365 ± 0.010	0.691 ± 0.018	0.564 ± 0.012
scGPT	0.753 ± 0.009	0.342 ± 0.007	0.280 ± 0.007	0.701 ± 0.012	0.519 ± 0.009
scGPT + *CellRefine* (Ours)	0.840 ± 0.008	0.377 ± 0.011	0.369 ± 0.006	0.740 ± 0.008	0.582 ± 0.008
Geneformer	0.731 ± 0.008	0.354 ± 0.006	0.284 ± 0.011	0.719 ± 0.009	0.522 ± 0.009
Geneformer + *CellRefine* (Ours)	$\mathbf{0.843 \pm 0.005}$	$\mathbf{0.391 \pm 0.006}$	$\mathbf{0.382 \pm 0.008}$	$\mathbf{0.754 \pm 0.003}$	$\mathbf{0.593 \pm 0.006}$

Table 1: On-domain cell identity prediction performance measured by macro-F1 (higher is better). Results are reported as mean ± standard deviation. Results of foundation models after applying *CellRefine* post-pretraining are highlighted in purple.

3.2 Baselines

We evaluated *CellRefine* against nine baselines for cell identity prediction, seven for spatial transcriptomics imputation, and three for zero-shot gene perturbation response prediction.

4 Results

In this section, we present the results of applying *CellRefine* to the three aforementioned tasks. We also provide an ablation study examining the contribution of each component of *CellRefine*, along with a qualitative analysis of the factors that may underlie its success in alignment.

4.1 Cell Identity Prediction

In Table 1, we report on-domain cell identity prediction results, comparing *CellRefine* against a range of baselines across four datasets. Results for the remaining datasets, and cell class-level analysis are provided in Appendix D.1 and Appendix D.2 respectively. Several findings stand out. First, post-pretraining with *CellRefine* consistently improves macro-F1 across all evaluated foundation-model backbones and datasets. In particular, *CellRefine* yields relative gains of more than 15% over the corresponding Geneformer baseline on the Blood and Liver datasets, both of which exhibit pronounced class imbalance and substantial cross-dataset generalization challenges. Overall, the strongest non-foundation-model baseline, scTOP representations on HVGs with logistic regression, achieves an average macro-F1 of 0.532. Although all three foundation-model baselines initially fall below this level, post-pretraining with *CellRefine* raises the performance of scBERT, scGPT, and Geneformer beyond it. Among these variants, Geneformer + *CellRefine* achieves the best overall performance, with an average macro-F1 of 0.593.

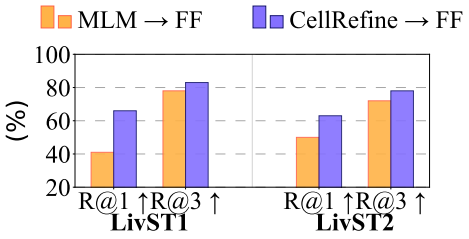


Figure 3: Out-of-domain zero-shot cell identity prediction performance (recall@k) on LivST1 and LivST2.

Next, we examine out-of-domain cell identity prediction. Here, models are trained on scRNA-seq data as the source domain and evaluated zero-shot on spatial transcriptomics data as the target domain. Although spatial transcriptomics provides gene expression measurements together with the spatial coordinates of cells within tissue, labeled datasets in this modality remain limited. As a result, models must generalize across distribution shifts between sequencing-based and spatial measurement technologies. We therefore assess whether *CellRefine*, trained on scRNA-seq data, can improve cell identity prediction in spatial transcriptomics data without target-domain supervision.

Since Geneformer is the strongest-performing foundation-model baseline, we compare it with Geneformer + *CellRefine*. Recall@k for $k \in \{1, 3\}$ is reported in Figure 3, where MLM → FF denotes

Method	LiverST1		LiverST2		Avg.	
	Cos ↑	Corr ↑	Cos ↑	Corr ↑	Cos ↑	Corr ↑
SpaGE	0.331 ± 0.012	0.192 ± 0.018	0.368 ± 0.009	0.160 ± 0.010	0.350 ± 0.011	0.176 ± 0.014
stPlus	0.382 ± 0.014	0.172 ± 0.011	0.291 ± 0.009	0.172 ± 0.016	0.337 ± 0.012	0.172 ± 0.014
Tangram	0.263 ± 0.005	0.114 ± 0.007	0.301 ± 0.008	0.166 ± 0.010	0.282 ± 0.007	0.140 ± 0.009
gimVI	0.312 ± 0.010	0.143 ± 0.011	0.327 ± 0.010	0.171 ± 0.009	0.320 ± 0.010	0.157 ± 0.010
scBERT	0.392 ± 0.012	0.288 ± 0.013	0.421 ± 0.019	0.261 ± 0.009	0.407 ± 0.016	0.275 ± 0.011
scBERT + *CellRefine* (Ours)	0.502 ± 0.016	0.300 ± 0.012	0.487 ± 0.017	0.289 ± 0.012	0.495 ± 0.017	0.295 ± 0.012
scGPT	0.421 ± 0.011	0.291 ± 0.009	0.442 ± 0.013	0.310 ± 0.016	0.432 ± 0.012	0.301 ± 0.013
scGPT + *CellRefine* (Ours)	0.494 ± 0.013	0.311 ± 0.009	0.502 ± 0.011	0.330 ± 0.011	0.498 ± 0.012	0.321 ± 0.010
Geneformer	0.477 ± 0.011	0.303 ± 0.010	0.491 ± 0.012	0.322 ± 0.009	0.484 ± 0.012	0.313 ± 0.010
Geneformer + *CellRefine* (Ours)	$\mathbf{0.526 \pm 0.013}$	$\mathbf{0.313 \pm 0.011}$	$\mathbf{0.514 \pm 0.008}$	$\mathbf{0.342 \pm 0.010}$	$\mathbf{0.520 \pm 0.011}$	$\mathbf{0.328 \pm 0.011}$

Table 2: Spatial transcriptomics imputation performance measured by cosine similarity (Cos) and Pearson correlation (Corr), higher is better. Results are reported as mean ± standard deviation.

Method	CD4 T	CD14+ Mono	CD8 T	Dendritic	Avg. ↑
scBERT	0.198 ± 0.003	0.518 ± 0.005	0.228 ± 0.007	0.421 ± 0.004	0.341 ± 0.005
scBERT + *CellRefine* (Ours)	0.214 ± 0.003	0.530 ± 0.002	0.232 ± 0.006	0.467 ± 0.002	0.361 ± 0.003
scGPT	0.210 ± 0.006	0.525 ± 0.004	0.229 ± 0.002	0.450 ± 0.006	0.354 ± 0.005
scGPT + *CellRefine* (Ours)	$\mathbf{0.234 \pm 0.008}$	0.539 ± 0.003	0.243 ± 0.002	0.481 ± 0.007	0.374 ± 0.005
Geneformer	0.204 ± 0.002	0.527 ± 0.003	0.231 ± 0.001	0.452 ± 0.001	0.354 ± 0.002
Geneformer + *CellRefine* (Ours)	0.232 ± 0.002	$\mathbf{0.542 \pm 0.001}$	$\mathbf{0.245 \pm 0.001}$	$\mathbf{0.485 \pm 0.001}$	$\mathbf{0.376 \pm 0.001}$

Table 3: Zero-shot gene perturbation response prediction results. Variants of *CellRefine* are highlighted in purple, and the best overall result for each dataset is shown in bold.

Geneformer and *CellRefine* → FF denotes Geneformer + *CellRefine*. Geneformer + *CellRefine* consistently demonstrates stronger zero-shot generalization across both datasets, with notable gains in Recall@1. These results suggest that *CellRefine* improves the latent cell embedding space, yielding more robust representations that may reduce noise and enhance transfer across modalities.

4.2 Spatial Transcriptomics Imputation

For the spatial transcriptomics imputation task, we evaluate four non-foundation-model baselines and three foundation models, scBERT, scGPT, and Geneformer, along with their corresponding variants post-pretrained with *CellRefine*. The foundation-model baselines generally outperform the non-foundation-model methods, with Geneformer achieving the strongest baseline performance. Consistent with the trend observed in cell identity prediction, post-pretraining with *CellRefine* further improves all three foundation-model backbones. The results are reported in Table 2.

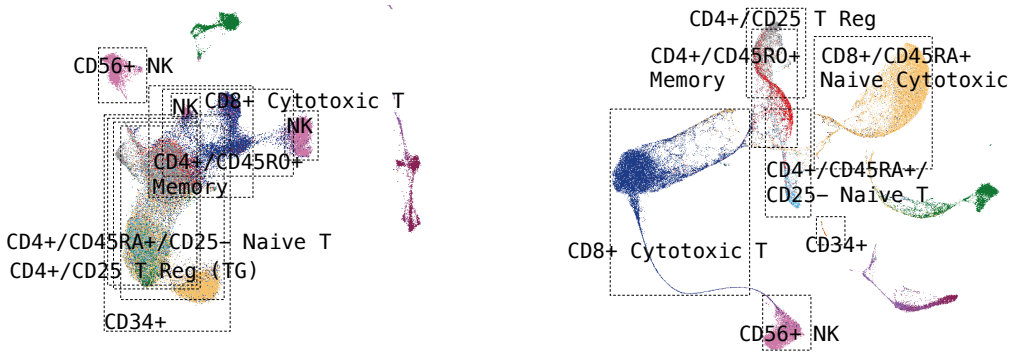
4.3 Gene Perturbation Response Prediction

The results for gene perturbation response prediction are shown in Table 3. This task is widely regarded as challenging, and prior studies have reported relatively low performance overall Kendiukhov (2026). Despite this difficulty, *CellRefine* yields consistent improvements over all three initial foundation model baselines, with scGPT + *CellRefine* leading in the cell type CD4 T and Geneformer + *CellRefine* achieving the overall strongest performance.

Additional result comparisons with parameter-efficient methods are provided in Appendix D.3.

4.4 Why *CellRefine* Works?

In this section, we qualitatively analyze the latent embedding space of Geneformer, which shows the strongest performance, before and after applying *CellRefine*, with the goal of gaining insight into the changes induced by *CellRefine* and why these changes may lead to improved overall performance, specifically with respect to the rare cells in the tail of the cell distribution. To this end, we use the Blood dataset Zheng et al. (2017) and extract latent cell embeddings both before and after applying *CellRefine*. We then use UMAP to project these embeddings into a two-dimensional space for visualization. The resulting plots are shown in Figure 4a (before applying *CellRefine*) and Figure 4b (after applying *CellRefine*), with cell clusters annotated with dashed lines.



(a) UMAP visualization of blood cell embeddings after applying MLM without *CellRefine* for alignment. Silhouette: 0.445, ARI: 0.258

(b) UMAP visualization of blood cell embeddings after post-pretraining with *CellRefine* for alignment. Silhouette: 0.695, ARI: 0.611

Figure 4: UMAP visualizations of latent Geneformer cell embeddings on the Blood dataset Zheng et al. (2017) before (Figure 4a) and after (Figure 4b) post-pretraining with *CellRefine*. Cell-type clusters are outlined with dashed lines. After applying *CellRefine*, the embeddings exhibit improved separation between cell clusters compared with the original Geneformer representations. The number of neighbors and the minimum distance used in generating UMAPs are 15 and 0.1, respectively.

Figures 4a and 4b suggest that *CellRefine* substantially reorganizes the latent embedding space, making underrepresented cell types more distinguishable. Prior to applying *CellRefine*, as shown in Figure 4a, tail cells exhibit substantial overlap with major cell populations. After refinement, Figure 4b shows distinct cell regions with clearer boundaries. This indicates that *CellRefine* promotes a more structured representation space in which underrepresented cell types are less obscured by dominant populations, better supporting downstream fine-tuning and explaining improvements across the three benchmark tasks. Additional visualizations of other datasets are given in Appendix D.7.

4.5 Effectiveness of the Multi-Term Objective

As described in Section 2.2.4, *CellRefine* optimizes four loss terms that jointly shape the learned representation space. In this section, we perform an ablation study to evaluate the contribution of each term to the overall performance by progressively adding the loss terms one at a time. In Table 4, we report average cell classification performance of the datasets in Table 1, average spatial transcriptomics imputation on liver spatial transcriptomic datasets in Table 2, and average perturbation response prediction across cell types in the perturbed peripheral blood dataset in Table 3. In all ablation experiments, we use Geneformer as the backbone, which is the strongest-performing foundation model in our evaluations. The progressive gains observed as additional objectives are introduced suggest that the loss terms provide complementary learning signals during post-pretraining. As the table indicates, *CellRefine* achieves the best performance when all four loss terms are included, confirming that each component of the multi-objective contributes to improving the embedding space and that their combination is particularly effective.

Loss Terms Used	CIP	STI	PRP
$\mathcal{L}_{\text{pre}}$	0.522	0.484	0.354
$\mathcal{L}_{\text{pre}} + \mathcal{L}_{C \rightleftarrows P}$	0.568	0.501	0.362
$\mathcal{L}_{\text{pre}} + \mathcal{L}_{C \rightleftarrows P} + \mathcal{L}_{\text{lineage}}$	0.572	0.511	0.368
$\mathcal{L}_{\text{pre}} + \mathcal{L}_{C \rightleftarrows P} + \mathcal{L}_{\text{lineage}} + \mathcal{L}_{\text{GMVE}}$	0.593	0.520	0.376

Table 4: Ablation study on the effectiveness of each loss term in *CellRefine* for cell identity prediction (CIP) (macro F1), spatial transcriptomics imputation (STI) (cosine similarity), and perturbation response prediction (PRP) (Pearson correlation coefficient).

Additional ablation experiments isolating the contribution of marker biology are provided in Appendix D.4. We further report a sensitivity analysis of the loss-function hyperparameters in Appendix D.5 and a comparison of computational overhead in Appendix D.6.

5 Related Work

In this section, we review related work, organized along a few key areas of interest.

Adaptation between pretraining and fine-tuning: Inserting a second pretraining stage before task-specific fine-tuning is established practice. Domain and task-adaptive pretraining continue the masked language modeling objective on in-domain data Gururangan et al. (2020), and post-pretraining has recently been used to realign vision–language models prior to downstream adaptation Yamaguchi et al. (2025). Although *CellRefine*also introduces an intermediate stage, the role of that stage is different. Continued MLM reuses the stage-one objective, which moves the embedding manifold toward the target distribution but imposes no structure the original objective did not already encode. *CellRefine*modifies the objective itself, adding terms defined over external biological structure rather than over observed tokens.

Prototype and ontology-guided representation learning: Anchoring representations to class-level embeddings is an established alternative to a free classification head, as in prototypical networks Snell et al. (2017) and in contrastive image–text pretraining, where a text encoder supplies the class anchors Radford et al. (2021). In single-cell analysis this has been realized through textual descriptions of cell types, as in LangCell Zhao et al. (2024), GenePT Chen & Zou (2025), and Cell2Sentence Levine et al. (2024). *CellRefine*differs in both the anchor and the stage at which it is applied. The anchor is an ontology-ordered marker gene program $\mathcal{H}(c_i)$ rather than free text, expressed in a form the pretrained tokenizer accepts directly and carrying the discriminative signal practitioners already use for manual annotation Dumitrascu et al. (2021). Both the prototype and lineage terms act as post-pretraining regularizers rather than classification objectives, so they constrain the representation without committing to a downstream head. In this sense, *CellRefine*is agnostic to the fine-tuning method that follows.

Long-tailed and shift-robust learning: Long-tailed recognition is typically addressed by reweighting or margin adjustment Cui et al. (2019); Cao et al. (2019), by logit adjustment under the class prior Menon et al. (2020), or by decoupling representation learning from classifier learning Kang et al. (2019). Within single-cell analysis, adaptive resampling has been proposed for imbalanced datasets Navidi et al. (2025). Covariate shift is usually treated separately, through batch integration methods such as Harmony Korsunsky et al. (2019) and scANVI Xu et al. (2021). *CellRefine*differs from both families in what it modifies. Reweighting and resampling methods require labeled tail examples and act primarily on the classifier, leaving the representation that classifier reads largely unchanged, while batch integration methods operate on expression values without access to cell-type semantics. *CellRefine*instead reshapes the representation using structure that exists for rare cell types even when samples do not, since marker programs and ontology relations are curated independently of any dataset. The same mechanism addresses both failure modes, because a marker-derived prototype is a target that neither depends on cell-type frequency nor varies with technical effects that leave marker identity intact.

6 Conclusion and Future Work

In this work, we introduce *CellRefine*, a post-pretraining approach for improving single-cell pretrained models prior to fine-tuning. Unlike conventional pipelines that rely solely on pretraining followed by task-specific adaptation, *CellRefine* incorporates biological priors during an intermediate optimization stage, refining cell embeddings and improving robustness to long-tail induced data imbalance and covariate shift. Across diverse computational biology tasks, *CellRefine* consistently improves downstream performance.

These results position *CellRefine* as a promising direction for single-cell representation learning and highlight the value of integrating biological domain knowledge into large-scale representation learning frameworks. A limitation of the current approach is its reliance on established marker knowledge, which makes it less suitable for novel or poorly characterized cell types. Future work can therefore explore ways to reduce this dependence and extend biologically guided post-pretraining to settings involving previously uncharacterized populations. The intended regime of *CellRefine* is instead cell types that are well characterized at the marker level but remain difficult to model because sufficiently large, high-quality labeled datasets are hard to obtain. Such difficulty can arise from rare biology, labeling ambiguity, and technical artifacts, all of which are largely orthogonal to whether marker knowledge is available. This is precisely the long-tailed and closely related cell-type regime in which the gains from *CellRefine* are most pronounced.

AI USE STATEMENT

In this work, we used generative AI tools for parts of the method implementation, primarily to edit software code. We have not used generative AI tools to help develop theoretical models or conceptual frameworks, formulate mathematical claims, propose or refine hypotheses, design or provide feedback on research methodology or experiments, implement methods (full implementation), assist with translation, clean and reformat dataset, support qualitative and thematic data analysis, interpret results, and generating synthetic data sets, providing critical ingredients for proving mathematical claims, assisting in the writing of proofs are not applicable to this work. Additionally, we used generative AI tools to edit the research paper to improve readability, format tables and figures, edit software code, identify relevant literature, and searching for information. We have reviewed all AI-assisted work. All LLM-generated software code was verified and tested for correctness by 2 authors. We conducted manual literature survey of LLM-suggested related work in the domain, and checked all figures and tables formatted by LLMs to make sure the contents were not distorted during formatting. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work does not involve human-subject recruitment, personally identifiable information, or the release of new datasets. All experiments are conducted using existing publicly available benchmark datasets and computational methods. We do not identify any additional ethical concerns specific to this work.

REPRODUCIBILITY STATEMENT

We provide detailed descriptions of the model architecture, training objectives, experimental setup, datasets, evaluation protocols, and hyperparameters in the main paper and appendix to facilitate reproducibility. Additional implementation details and ablation settings are also included in the appendix. We will publicly release the source code and necessary scripts upon acceptance.

REFERENCES

10X Genomics. Human liver data, xenium human multi-tissue and cancer panel, November 2023. Single Cell Spatial Dataset by Xenium Analyzer v2.0.

10x Genomics. 10x Genomics Datasets. `https://www.10xgenomics.com/datasets`, 2025. Accessed: 2025-11-20.

Elham Azizi, Ambrose J Carr, George Plitas, Andrew E Cornish, Catherine Konopacki, Sandhya Prabhakaran, Juozas Nainys, Kenmin Wu, Vaidotas Kiseliovas, Manu Setty, et al. Single-cell map of diverse immune phenotypes in the breast tumor microenvironment. *Cell*, 174(5):1293–1308, 2018.

Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. *Advances in neural information processing systems*, 32, 2019.

Jiawei Chen, Hao Xu, Wanyu Tao, Zhaoxiong Chen, Yuxuan Zhao, and Jing-Dong J Han. Transformer for one stop interpretable cell type annotation. *Nature Communications*, 14(1):223, 2023.

Yiqun Chen and James Zou. Simple and effective embedding model for single-cell biology built from chatgpt. *Nature biomedical engineering*, 9(4):483–493, 2025.

Sijin Cheng, Ziyi Li, Ranran Gao, Baocai Xing, Yunong Gao, Yu Yang, Shishang Qin, Lei Zhang, Hanqiang Ouyang, Peng Du, et al. A pan-cancer single-cell transcriptional atlas of tumor infiltrating myeloid cells. *Cell*, 184(3):792–809, 2021.

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

Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *2019 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 9260–9269. IEEE, 2019.

Nat Dilokthanakul, Pedro AM Mediano, Marta Garnelo, Matthew CH Lee, Hugh Salimbeni, Kai Arulkumaran, and Murray Shanahan. Deep unsupervised clustering with gaussian mixture variational autoencoders. *arXiv preprint arXiv:1611.02648*, 2016.

B Dumitrascu, S Villar, DG Mixon, and BE Engelhardt. Optimal marker gene selection for cell type discrimination in single cell analyses. nat. commun. 12, 1186, 2021.

Oscar Franzén, Li-Ming Gan, and Johan LM Björkegren. Panglaodb: a web server for exploration of mouse and human single-cell rna sequencing data. *Database*, 2019:baz046, 2019.

Adam Gayoso, Romain Lopez, Galen Xing, Pierre Boyeau, Valeh Valiollah Pour Amiri, Justin Hong, Katherine Wu, Michael Jayasuriya, Edouard Mehlman, Maxime Langevin, et al. A python library for probabilistic analysis of single-cell omics data. *Nature biotechnology*, 40(2):163–166, 2022.

Dominic Grün, Anna Lyubimova, Lennart Kester, Kay Wiebrands, Onur Basak, Nobuo Sasaki, Hans Clevers, and Alexander Van Oudenaarden. Single-cell messenger rna sequencing reveals rare intestinal cell types. *Nature*, 525(7568):251–255, 2015.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A Smith. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pp. 8342–8360, 2020.

Congxue Hu, Tengyue Li, Yingqi Xu, Xinxin Zhang, Feng Li, Jing Bai, Jing Chen, Wenqi Jiang, Kaiyue Yang, Qi Ou, et al. Cellmarker 2.0: an updated database of manually curated cell markers in human/mouse and web tools based on scrna-seq data. *Nucleic acids research*, 51(D1):D870–D876, 2023.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.

Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition. *arXiv preprint arXiv:1910.09217*, 2019.

Hyun Min Kang, Meena Subramaniam, Sasha Targ, Michelle Nguyen, Lenka Maliskova, Elizabeth McCarthy, Eunice Wan, Simon Wong, Lauren Byrnes, Cristina M Lanata, et al. Multiplexed droplet single-cell rna-sequencing using natural genetic variation. *Nature biotechnology*, 36(1): 89–94, 2018.

Ihor Kendiukhov. Sparse autoencoders reveal organized biological knowledge but minimal regulatory logic in single-cell foundation models: a comparative atlas of geneformer and scgpt. *arXiv preprint arXiv:2603.02952*, 2026.

Nayoung Kim, Hong Kwan Kim, Kyungjong Lee, Yourae Hong, Jong Ho Cho, Jung Won Choi, Jung-Il Lee, Yeon-Lim Suh, Bo Mi Ku, Hye Hyeon Eum, et al. Single-cell rna sequencing demonstrates the molecular and cellular reprogramming of metastatic lung adenocarcinoma. *Nature communications*, 11(1):2285, 2020.

Ilya Korsunsky, Nghia Millard, Jean Fan, Kamil Slowikowski, Fan Zhang, Kevin Wei, Yuriy Baglaenko, Michael Brenner, Po-ru Loh, and Soumya Raychaudhuri. Fast, sensitive and accurate integration of single-cell data with harmony. *Nature methods*, 16(12):1289–1296, 2019.

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, 2024.

Arthur Liberzon, Aravind Subramanian, Reid Pinchback, Helga Thorvaldsdóttir, Pablo Tamayo, and Jill P Mesirov. Molecular signatures database (msigdb) 3.0. *Bioinformatics*, 27(12):1739–1740, 2011.

Yingxin Lin, Yue Cao, Hani Jieun Kim, Agus Salim, Terence P Speed, David M Lin, Pengyi Yang, and Jean Yee Hwa Yang. scclassify: sample size estimation and multiscale classification of cells using single and multiple reference. *Molecular systems biology*, 16(6):MSB199389, 2020.

Aditya Krishna Menon, Sadeep Jayasumana, Ankit Singh Rawat, Himanshu Jain, Andreas Veit, and Sanjiv Kumar. Long-tail learning via logit adjustment. *arXiv preprint arXiv:2007.07314*, 2020.

Zeinab Navidi, Akshaya Thoutam, Madeline Hughes, Srivatsan Raghavan, Peter S Winter, Lorin Crawford, and Ava P Amini. Adaptive resampling for improved machine learning in imbalanced single-cell datasets. *bioRxiv*, 2025.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

Lucas Schirmer, Dmitry Velmeshev, Staffan Holmqvist, Max Kaufmann, Sebastian Werneburg, Diane Jung, Stephanie Vistnes, John H Stockley, Adam Young, Maike Steindel, et al. Neuronal vulnerability and multilineage diversity in multiple sclerosis. *Nature*, 573(7772):75–82, 2019.

Artem Shmatko, Alexander Wolfgang Jung, Kumar Gaurav, Søren Brunak, Laust Hvas Mortensen, Ewan Birney, Tom Fitzgerald, and Moritz Gerstung. Learning the natural history of human disease with generative transformers. *Nature*, pp. 1–9, 2025.

Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30, 2017.

Aravind Subramanian, Pablo Tamayo, Vamsi K Mootha, Sayan Mukherjee, Benjamin L Ebert, Michael A Gillette, Amanda Paulovich, Scott L Pomeroy, Todd R Golub, Eric S Lander, et al. Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. *Proceedings of the national academy of sciences*, 102(43):15545–15550, 2005.

Chin Leng Tan, K Lindner, T Boschert, Z Meng, A Rodriguez Ehrenfried, A De Roia, G Haltenhof, A Faenza, F Imperatore, L Bunse, et al. Prediction of tumor-reactive t cell receptors from scrna-seq data for personalized t cell therapy. *Nature biotechnology*, 43(1):134–142, 2025.

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.

Nathan R Tucker, Mark Chaffin, Stephen J Fleming, Amelia W Hall, Victoria A Parsons, Kenneth C Bedi Jr, Amer-Denis Akkad, Caroline N Herndon, Alessandro Arduini, Irinna Papangeli, et al. Transcriptional and cellular diversity of the human heart. *Circulation*, 142(5):466–482, 2020.

Alexandra-Chloé Villani, Rahul Satija, Gary Reynolds, Siranush Sarkizova, Karthik Shekhar, James Fletcher, Morgane Griesbeck, Andrew Butler, Shiwei Zheng, Suzan Lazo, et al. Single-cell rna-seq reveals new types of human blood dendritic cells, monocytes, and progenitors. *Science*, 356 (6335):eaah4573, 2017.

Hongzhi Wen, Wenzhuo Tang, Xinnan Dai, Jiayuan Ding, Wei Jin, Yuying Xie, and Jiliang Tang. Cellplm: pre-training of cell language model beyond single cells. biorxiv. *preprint*, 2023.

Yang Xia, Yin Zhu, Sai Zhang, Mo Zhou, Jiaqi Zhou, Guanxin Xu, Baiqin Zhao, Guofei Zhang, Qiyuan Wang, Bin Wang, et al. Precise diagnosis of small invasive pulmonary nodules driven by single-cell immune signatures in peripheral blood. *Nature communications*, 16(1):10939, 2025.

Chenling Xu, Romain Lopez, Edouard Mehlman, Jeffrey Regier, Michael I Jordan, and Nir Yosef. Probabilistic harmonization and annotation of single-cell transcriptomics data with deep generative models. *Molecular systems biology*, 17(1):MSB20209620, 2021.

Shin'ya Yamaguchi, Dewei Feng, Sekitoshi Kanai, Kazuki Adachi, and Daiki Chijiwa. Post-pre-training for modality alignment in vision-language foundation models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4256–4266, 2025.

Suyuan Zhao, Jiahuan Zhang, Yushuai Wu, Yizhen Luo, and Zaiqing Nie. Langcell: Language-cell pre-training for cell identity understanding. *arXiv preprint arXiv:2405.06708*, 2024.

Grace XY Zheng, Jessica M Terry, Phillip Belgrader, Paul Ryvkin, Zachary W Bent, Ryan Wilson, Solongo B Ziraldo, Tobias D Wheeler, Geoff P McDermott, Junjie Zhu, et al. Massively parallel digital transcriptional profiling of single cells. *Nature communications*, 8(1):14049, 2017.

Appendix

A Broader Impacts **15**

B Gene marker set and prototype construction **15**

C SCRL as Language Prediction Tasks **16**

D Additional Experiments and Results **17**

D.1 Cell Identity Prediction Performance on Additional Benchmark Datasets 17

D.2 Class-level cell identity prediction performance 18

D.3 Experiments with additional parameter efficient baselines 19

D.4 Experiments Isolating the Contribution of Marker Biology 20

D.5 Loss Function Hyperparameter Sensitivity Analysis 21

D.6 Compute overhead comparison 22

D.7 Latent space qualitative analysis 22

E Experimental Settings **23**

F Dataset details **26**

G Evaluation Metrics **26**

G.1 Cell Type Annotation 26

G.2 Spatial Transcriptomics Imputation 27

G.3 Gene Perturbation Response Prediction 28

H Cell Ontologies and Type Distributions **28**

I Long tail quantification **28**

J Additional dataset details **29**

A BROADER IMPACTS

Advances in single-cell representation learning, such as the framework proposed in this work, have the potential to significantly accelerate progress in biomedical research and healthcare. By enabling more accurate and robust characterization of cellular identities, particularly for rare and clinically relevant cell populations, our approach may contribute to improved understanding of disease mechanisms, earlier detection of pathological states, and the development of more targeted therapeutic interventions. In particular, better modeling of rare cell types could facilitate breakthroughs in areas such as cancer biology, immunology, and regenerative medicine, where small but critical cell populations often drive disease progression or treatment response.

Beyond direct clinical applications, our method may also enhance the utility of large-scale single-cell atlases and public datasets by improving cross-dataset generalization in the presence of technical variability. This could promote more reproducible and integrative analyses across research groups, enabling more efficient knowledge sharing and collaborative discovery. Furthermore, incorporating biologically grounded inductive biases into machine learning models may encourage a broader shift toward hybrid approaches that combine data-driven learning with domain expertise, potentially improving interpretability and trustworthiness in computational biology.

However, several potential risks and limitations should be considered. First, while our approach aims to better capture rare cell populations, biases in the underlying datasets—such as underrepresentation of certain populations, disease states, or demographic groups—may still propagate into the learned representations. This could lead to uneven performance across biological contexts and limit the generalizability of the model in real-world clinical settings. Second, reliance on curated biological knowledge, such as marker gene databases and cell ontologies, introduces dependencies on the completeness and accuracy of these resources. Errors or omissions in these databases could bias the model or reinforce existing misconceptions in the literature.

Additionally, as with many machine learning models applied to biomedical data, there are considerations related to data privacy and ethical use. Although our work focuses on aggregated gene expression data rather than identifiable patient information, downstream applications in clinical settings may involve sensitive datasets, requiring careful handling, appropriate consent, and adherence to regulatory standards.

Finally, while improved predictive performance can support scientific discovery, such models should not be used in isolation for clinical decision-making without rigorous validation. Ensuring transparency, interpretability, and proper benchmarking across diverse datasets will be essential for responsible deployment.

Overall, this work contributes toward more biologically informed and robust machine learning methods for single-cell analysis, with the potential for meaningful positive impact in biomedical research and healthcare, while highlighting the importance of careful consideration of data biases, knowledge dependencies, and ethical use.

B GENE MARKER SET AND PROTOTYPE CONSTRUCTION

For each cell in the post-pretraining cohort, we construct a comprehensive set of marker genes by aggregating curated human cell type markers from multiple high-quality biological databases. Specifically, we integrate marker gene information from the MSigDB C8 collection Subramanian et al. (2005); Liberzon et al. (2011), CellMarker 2.0 Hu et al. (2023), and PanglaoDB Franzén et al. (2019). These resources provide complementary coverage of cell type–specific gene signatures across a wide range of tissues and biological contexts. By combining these sources, we obtain a robust and diverse marker gene set for each cell type, reducing reliance on any single database and improving coverage of both well-characterized and less-studied cell populations.

Once the marker gene sets are compiled, we organize them according to their positions within the cell ontology using a marker organization function $H(c_i)$, where c_i denotes a specific cell type. The purpose of this function is to impose a biologically meaningful ordering on the marker genes that reflects the hierarchical structure of the cell ontology. Rather than treating marker genes as an unordered set, $H(c_i)$ leverages the ontology to encode relationships between cell types at different levels of granularity.

In particular, $H(c_i)$ orders marker genes based on their specificity within the ontology hierarchy. Genes associated with highly specific cell identities (e.g., fine-grained sub-subtypes) are given higher priority and placed earlier in the sequence. These are followed by genes characteristic of intermediate cell subtypes, and finally by genes associated with broader, high-level cell categories. This ordering captures a progression from fine-grained to coarse-grained biological identity, enabling the representation to emphasize discriminative features of closely related cell types while still retaining information about broader lineage relationships.

For example, consider a marker gene set $\{A, B, C\}$ for a given cell type, where gene A is associated with a major cell type, gene B corresponds to an intermediate cell subtype, and gene C is specific to a more refined cell sub-subtype. Applying the organization function $H(c_i)$ reorders this set into the sequence (C, B, A), reflecting a hierarchy from the most specific to the most general markers.

We refer to this ordered sequence of marker genes as the *prototype* of the cell type. This prototype serves as a structured representation that encapsulates both the defining molecular features of the cell and their hierarchical relationships within the broader cell ontology.

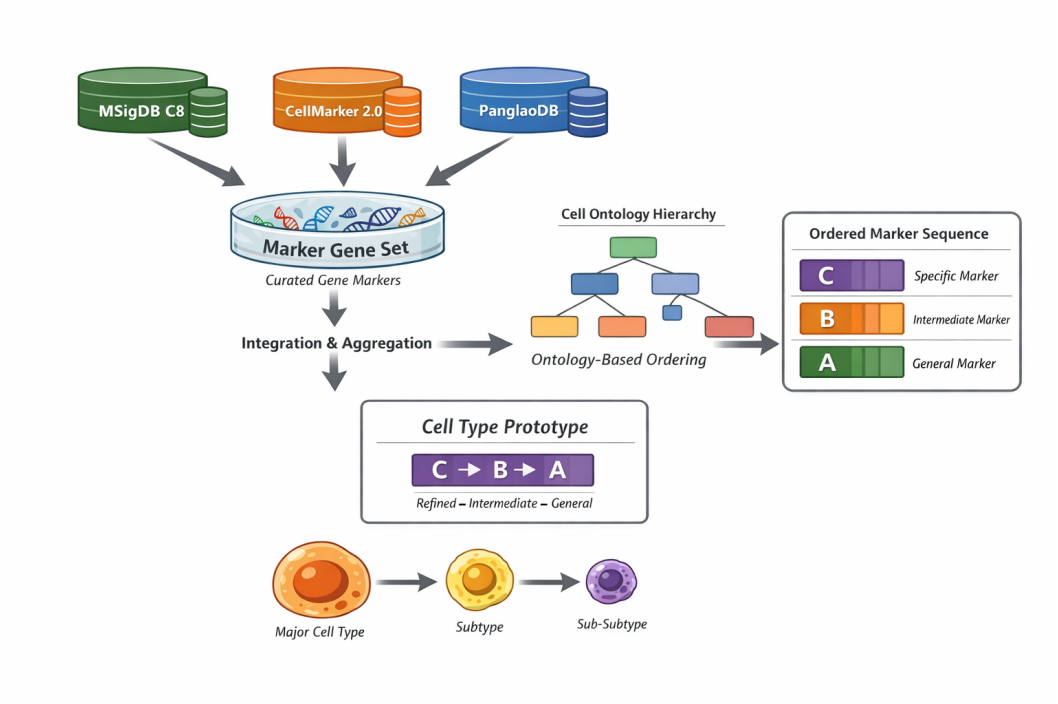


Figure 5: Marker gene set and cell type prototype creation procedure.

C SCRL AS LANGUAGE PREDICTION TASKS

Many pretrained models proposed for SCRL employ self-supervised token prediction objectives, including masked language modeling and autoregressive-style prediction Theodoris et al. (2023). These objectives provide a natural mechanism for learning dependencies among genes and their expression patterns from large collections of unlabeled single-cell data. In this section, we formulate these two common pretraining paradigms and describe their application to SCRL.

We begin by representing each cell as a sequence of gene tokens drawn from a fixed vocabulary $\mathcal{V} = \{x_1, \ldots, x_k\}$. For each cell c, a deterministic tokenization procedure $\tau(\cdot)$ maps its gene expression profile into an ordered token sequence of length t,

$$\mathbf{c} = \tau(c) = (x_1, \ldots, x_t), \qquad x_i \in \mathcal{V}. \tag{9}$$

Masked language modeling. In masked language modeling, a subset of token indices $M \subseteq \{1, \ldots, t\}$ is masked from the input. Let

$$\mathbf{c}_{\mathrm{o}} = (x_i : i \notin M) \tag{10}$$

denote the observed tokens. The objective is to learn a parameterized model f_θ that captures the conditional distribution of each masked token given the observed context,

$$f_\theta : \mathbf{c}_{\mathrm{o}} \longmapsto p_\theta(x_i \mid \mathbf{c}_{\mathrm{o}}), \qquad i \in M. \tag{11}$$

The model parameters θ are optimized by minimizing the masked-token-modeling objective

$$\mathcal{L}_{\mathrm{MLM}} = -\sum_{i \in M} \log p_\theta(x_i \mid \mathbf{c}_{\mathrm{o}}) + \alpha \mathcal{L}(\theta), \tag{12}$$

where $\mathcal{L}(\theta)$ denotes an optional regularization term and α controls its contribution to the objective. In our experiments, we use $\mathcal{L}_{\mathrm{MLM}}$ as $\mathcal{L}_{\mathrm{pre}}$ for scBERT and Geneformer, following their original pretraining schemes.

Autoregressive token prediction. An alternative is to learn the cell representation through autoregressive prediction. Given an ordered token sequence $\mathbf{c} = (x_1, \ldots, x_t)$, the joint probability of the sequence can be factorized as

$$p_\theta(\mathbf{c}) = \prod_{i=1}^{t} p_\theta(x_i \mid x_{<i}), \tag{13}$$

where $x_{<i} = (x_1, \ldots, x_{i-1})$ denotes the tokens preceding position i. The model is therefore trained to predict each token conditioned on the previously observed tokens,

$$f_\theta : x_{<i} \longmapsto p_\theta(x_i \mid x_{<i}). \tag{14}$$

The corresponding autoregressive objective is

$$\mathcal{L}_{\mathrm{AR}} = -\sum_{i=1}^{t} \log p_\theta(x_i \mid x_{<i}) + \alpha \mathcal{L}(\theta). \tag{15}$$

In our experiments, we use $\mathcal{L}_{\mathrm{AR}}$ as $\mathcal{L}_{\mathrm{pre}}$ for scGPT, following it's original pretraining schemes.

Variants of this formulation are used by generative single-cell foundation models, where specialized attention masks can define which genes or expression values are treated as observed and which are predicted. Thus, although the precise prediction target and masking strategy may differ across models, both masked and autoregressive objectives train the model by recovering cellular information from partial context.

In both cases, the self-supervised prediction objective encourages the model to capture statistical dependencies among genes and gene-expression patterns. The resulting internal representations can subsequently be adapted to downstream SCRL tasks such as cell identity prediction, spatial transcriptomics imputation, and perturbation-response prediction.

D Additional Experiments and Results

D.1 Cell Identity Prediction Performance on Additional Benchmark Datasets

In Table 5, we report cell identity prediction performance on three additional benchmark datasets: Pancreas, PBMC10K, and Heart. Among the baselines, logistic regression applied to scTOP representations on HVGs achieves the strongest overall performance, while all three foundation-model

Method	Pancreas	PBMC10K	Heart	Overall ↑
PCA + LR	0.809 ± 0.009	0.954 ± 0.007	0.871 ± 0.005	0.878 ± 0.007
LR on HVG	0.822 ± 0.003	0.953 ± 0.007	0.803 ± 0.005	0.859 ± 0.005
scTOP	0.842 ± 0.000	0.924 ± 0.014	0.846 ± 0.004	0.871 ± 0.006
scTOP-rep on HVG + LR	0.826 ± 0.006	0.954 ± 0.004	0.887 ± 0.004	0.889 ± 0.005
scANVI	0.704 ± 0.007	0.902 ± 0.003	0.859 ± 0.006	0.822 ± 0.005
CellTypist	0.713 ± 0.001	0.886 ± 0.009	0.864 ± 0.004	0.821 ± 0.005
scBERT	0.692 ± 0.007	0.915 ± 0.004	0.841 ± 0.007	0.816 ± 0.006
scBERT + CellRefine	0.811 ± 0.007	0.946 ± 0.007	0.906 ± 0.011	0.888 ± 0.008
scGPT	0.716 ± 0.010	0.942 ± 0.003	0.849 ± 0.010	0.836 ± 0.008
scGPT + CellRefine	$\mathbf{0.842 \pm 0.007}$	0.953 ± 0.014	0.907 ± 0.007	$\mathbf{0.901 \pm 0.009}$
Geneformer	0.721 ± 0.007	0.951 ± 0.007	0.865 ± 0.006	0.846 ± 0.007
Geneformer + CellRefine	0.829 ± 0.005	$\mathbf{0.963 \pm 0.007}$	$\mathbf{0.912 \pm 0.005}$	$\mathbf{0.901 \pm 0.006}$

Table 5: Cell identity prediction performance measured by macro-F1 (higher is better). Results are reported as mean ± standard deviation.

Cell type	Cells	scTOP-rep on HVG + LR	Geneformer	CellRefine → FF	Δ vs. backbone
CD8+ Cytotoxic T	20,757	0.90	0.914	0.941	+0.027
CD8+/CD45RA+ Naive Cytotoxic	16,645	0.89	0.848	0.947	+0.099
CD56+ NK	8,775	0.94	0.896	0.965	+0.069
CD4+/CD25 T Reg	6,185	0.77	0.792	0.860	+0.068
CD19+ B	5,877	0.79	0.841	0.934	+0.093
CD4+/CD45RO+ Memory	3,059	0.70	0.669	0.800	+0.131
CD14+ Monocyte	2,847	0.86	0.845	0.924	+0.079
Dendritic	2,095	0.75	0.708	0.836	+0.128
CD4+/CD45RA+/CD25$^-$ Naive T	1,871	0.63	0.500	0.781	+0.281
CD34+	242	0.84	0.924	0.965	+0.041
CD4+ T Helper2	97	0.06	0.126	0.320	+0.194
Macro-F1		**0.739**	**0.733**	**0.843**	**+0.110**

Table 6: Class-level cell identity prediction performance on the Blood dataset. *CellRefine* is applied to the Geneformer backbone. Δ denotes the improvement over the Geneformer backbone.

baselines initially fall below it. Post-pretraining with *CellRefine*, however, consistently improves the performance of each foundation-model backbone, enabling all three to surpass the strongest baseline. This trend is consistent with the results reported on the benchmark datasets in the main paper.

D.2 Class-level cell identity prediction performance

Table 6 presents class-level cell identity prediction performance. We draw below observations.

(1) Post-pretraining improves every class, and CellRefine is the only method that dominates the classical baseline across the board. CellRefine raises F1 on all eleven cell types relative to the Geneformer backbone (macro 0.733→0.843, +15.0%), and it exceeds the strong classical baseline on all eleven, whereas the off-the-shelf backbone does not (Geneformer, at 0.733, is below the 0.739 classical baseline).

(2) The gains are concentrated on the transcriptionally confusable subtypes, where the off-the-shelf model is weakest. The abundant, distinct types (CD8+ Cytotoxic, NK, CD34+) are already near-ceiling and gain modestly (+0.03 to +0.07). The large gains fall on the closely related subsets that all methods confuse: CD4+ Naive T (0.500→0.781, +56%), CD4+ Memory (0.669→0.800), Dendritic (0.708→0.836), the CD8 naive/cytotoxic pair (0.848→0.947), and Treg (0.792→0.860). The CD4+ T Helper2 class improves (0.06→0.126→0.320) but still remains the hardest class.

Method	Blood	Pancreas	Myeloid	Liver	LivST1	LivST2	MS	Lung	PBMC10K	Heart	Avg. ↑
LP	0.14	0.12	0.14	0.11	0.13	0.11	0.09	0.16	0.19	0.13	0.13
LL	0.54	0.49	0.21	0.16	0.37	0.28	0.41	0.48	0.62	0.49	0.41
LoRA	0.60	0.68	0.30	0.18	0.50	0.51	0.68	0.89	0.93	0.77	0.60
FF	0.67	0.69	0.34	0.27	0.64	0.60	0.70	0.92	0.93	0.86	0.66
MLM LL → LL	0.55	0.56	0.20	0.16	0.42	0.31	0.40	0.51	0.61	0.47	0.42
MLM LoRA → LoRA	0.63	0.72	0.32	0.18	0.51	0.53	0.65	0.90	0.92	0.8	0.62
MLM → FF	0.73	0.72	0.35	0.28	0.68	0.61	0.72	0.95	0.95	0.86	0.69
CellRefine LL → LL (Ours)	0.68	0.74	0.24	0.16	0.40	0.41	0.44	0.49	0.64	0.51	0.47
CellRefine LoRA → LoRA (Ours)	0.79	0.82	0.37	0.31	0.71	0.70	0.71	0.93	0.95	0.85	0.71
CellRefine → FF (Ours)	**0.84**	**0.83**	**0.39**	**0.38**	**0.77**	**0.73**	**0.75**	**0.97**	**0.96**	**0.91**	**0.75**

Table 7: On-domain cell identity prediction performance across 10 benchmark datasets, measured by macro-F1 (higher is better). Variants of *CellRefine* are highlighted in purple, and the best overall result for each dataset is shown in bold.

Method	LiverST1		LiverST2	
	Cos ↑	Cor ↑	Cos ↑	Cor ↑
LoRA	0.43	0.28	0.42	0.29
FF	0.48	0.30	0.49	0.32
CellRefine LoRA → LoRA (Ours)	0.46	0.29	0.46	0.31
CellRefine → FF (Ours)	**0.52**	**0.31**	**0.51**	**0.34**

Table 8: Spatial transcriptomics imputation performance on LivST1 and LivST2. Best results are shown in bold.

D.3 Experiments with additional parameter efficient baselines

In addition to the baselines in section 3.2, we tested the following parameter-efficient variations of the strongest performing foundation model, Geneformer Theodoris et al. (2023), aiming to observe the learning efficiency.

Linear Probing (LP): freeze the pre-trained weights and train a linear layer with the task loss. **Last-Layer Fine-Tuning (LL)**: fine-tune only the final layer of the foundation model with the task loss. **Full Fine-Tuning (FF)**: fine-tune the entire foundation model with the task loss. **MLM LL → LL**: first perform last-layer MLM post-training with the given dataset and then perform last-layer fine-tuning with the task loss. **MLM → FF**: first perform continued MLM post-training with the given dataset and then perform full fine-tuning with the task loss. **LoRA**: perform parameter-efficient fine-tuning with a frozen backbone and trainable LoRA adapters of the foundational model Hu et al. (2022). **MLM LoRA → LoRA**: first perform continued MLM post-training updating only the LoRA adapter parameters, followed by LoRA fine-tuning with the task loss.

In addition to the baselines, we also define three variants of *CellRefine* for comparison. ***CellRefine* LL → LL**: perform post-pretraining with the given dataset using *CellRefine* but only update the last layer of the foundational model. ***CellRefine* LoRA → LoRA**: perform post-pretraining with the given dataset using *CellRefine* but use LoRA adapters. ***CellRefine* → FF**: perform post-pretraining with the given dataset using *CellRefine* and update the full foundational model.

Method	CD4 T	CD8 T	NK	B	CD14+ Mono	FCGR3A+ Mono	Dendritic	Avg. ↑
LoRA	0.18	0.20	0.22	0.24	0.49	0.48	**0.46**	0.32
FF	0.20	0.23	0.23	0.24	0.52	0.49	0.44	0.34
MLM LoRA → LoRA	0.18	0.21	0.22	0.23	0.50	0.48	0.45	0.32
MLM → FF	0.20	0.23	**0.24**	0.25	0.53	**0.50**	0.45	0.34
CellRefine LoRA → LoRA (Ours)	**0.23**	0.23	0.23	0.24	**0.54**	0.49	**0.46**	0.34
CellRefine → FF (Ours)	**0.23**	**0.25**	**0.24**	**0.26**	**0.54**	**0.50**	**0.49**	**0.36**

Table 9: Zero-shot gene perturbation response prediction results. Variants of *CellRefine* are highlighted in purple, and the best overall result for each dataset is shown in bold.

In addition to Table 1 in the main paper, here we report extended results on on-domain cell identity prediction results in Table 7, comparing *CellRefine* with the set of baselines across 10 datasets. *CellRefine* consistently achieves stronger performance in terms of macro-F1 across all datasets, out-

Method	Blood	Liver	Myeloid	Avg. ↑
MLM → FF	0.731 ± 0.008	0.284 ± 0.011	0.354 ± 0.006	0.456 ± 0.008
(1) Compute-matched MLM → FF	0.729 ± 0.007	0.291 ± 0.006	0.347 ± 0.009	0.456 ± 0.007
(2) Supervised contrastive, no markers	0.722 ± 0.010	0.274 ± 0.009	0.350 ± 0.008	0.449 ± 0.009
(3) Randomly initialized learnable prototype alignment → FF	0.751 ± 0.004	0.343 ± 0.005	0.366 ± 0.005	0.487 ± 0.005
(4) Random marker alignment → FF	0.742 ± 0.009	0.334 ± 0.007	0.351 ± 0.009	0.476 ± 0.008
(5) Shuffled marker alignment → FF	0.789 ± 0.009	0.345 ± 0.008	0.371 ± 0.012	0.502 ± 0.010
(6) 25% markers	0.745 ± 0.007	0.362 ± 0.007	0.363 ± 0.003	0.490 ± 0.006
(7) 50% markers	0.773 ± 0.006	$\mathbf{0.382 \pm 0.004}$	0.364 ± 0.003	0.506 ± 0.004
(8) 75% markers	0.820 ± 0.004	0.380 ± 0.004	0.381 ± 0.007	0.527 ± 0.005
CellRefine → FF	$\mathbf{0.843 \pm 0.005}$	$\mathbf{0.382 \pm 0.008}$	$\mathbf{0.391 \pm 0.006}$	$\mathbf{0.539 \pm 0.006}$

Table 10: Ablation study isolating the contribution of marker biology in cell identity prediction. Results are reported as mean ± standard deviation in macro-F1, where higher is better.

Method	LiverST1	LiverST2	Avg. ↑
MLM → FF	0.477 ± 0.011	0.491 ± 0.012	0.484 ± 0.012
(1) Compute-matched MLM → FF	0.492 ± 0.008	0.491 ± 0.007	0.492 ± 0.008
(2) Supervised contrastive, no markers	0.475 ± 0.008	0.476 ± 0.013	0.476 ± 0.011
(3) Randomly initialized learnable prototype alignment → FF	0.490 ± 0.012	0.486 ± 0.008	0.488 ± 0.010
(4) Random marker alignment → FF	0.481 ± 0.009	0.483 ± 0.011	0.482 ± 0.010
(5) 25% markers	0.492 ± 0.011	0.494 ± 0.008	0.493 ± 0.010
(6) 50% markers	0.496 ± 0.007	0.501 ± 0.009	0.499 ± 0.008
(7) 75% markers	0.514 ± 0.008	0.508 ± 0.014	0.511 ± 0.011
CellRefine → FF	$\mathbf{0.526 \pm 0.013}$	$\mathbf{0.514 \pm 0.008}$	$\mathbf{0.520 \pm 0.011}$

Table 11: Ablation isolating the contribution of marker biology for spatial transcriptomics imputation. Results are reported as mean ± standard deviation in cosine similarity, where higher is better.

performing existing alignment methods. Notable performance gains of over 15% relative to the strongest baseline MLM → FF are observed on Blood, Pancreas, Liver, LivST1, and LivST2, which exhibit pronounced class imbalance and substantial cross-dataset generalization challenges. Overall, Full fine-tuning (FF) achieves an average macro-F1 of 0.66, while MLM alignment followed by full fine-tuning (MLM → FF) improves this to 0.69. In contrast, full fine-tuning with CellRefine (*CellRefine* → FF) achieves a substantially higher macro-F1 of 0.75. These improvements are consistent across fine-tuning strategies, including full and last-layer fine-tuning, and LoRA, which are strong evidence of the efficiency and effectiveness of *CellRefine*.

In the spatial transcriptomics imputation task, we observe a trend similar to that in the cell identity prediction task: *CellRefine* consistently outperforms the baselines. The results are presented in Table 8. We note that, because the objective in this task is to impute masked genes, it is naturally formulated as a masked language modeling problem. As a result, the only applicable baselines are FF and LoRA, both of which are trained using the masked language modeling loss.

Table 9 presents Pearson Correlation Coefficients of the zero-shot perturbation response prediction task. We note that when *CellRefine* post-pretraining is applied, LoRA variant outperforms the LoRA variant of Geneformer without *CellRefine* post-pretraining.

D.4 Experiments Isolating the Contribution of Marker Biology

In this section, we present the results isolating the contribution of marker biology. We used Geneformer backbone for all the runs presented in Table 10, Table 11, and Table 12. MLM → FF refers to the initial model prior to applying *CellRefine* post-pretraining, and *CellRefine* → FF refers to the model after applying *CellRefine* post-pretraining.

In the cell identity prediction task presented in Table 10, the compute-matched MLM → FF performs the same as the baseline. Supervised contrastive learning without markers is almost on par with the baseline, indicating that generic label supervision alone does not help. The content-free prototype controls, namely randomly initialized learnable prototypes and random-marker alignment, remain well below our method. As we increase the fraction of marker genes, macro-F1 rises gradually, with *CellRefine* (full markers) obtaining the best result of 0.539.

Method	CD4 T	CD8 T	Avg. ↑
MLM → FF	0.204 ± 0.002	0.231 ± 0.001	0.218 ± 0.001
(1) Compute-matched MLM → FF	0.208 ± 0.0009	0.225 ± 0.002	0.217 ± 0.001
(2) Supervised contrastive, no markers	0.200 ± 0.003	0.233 ± 0.002	0.217 ± 0.002
(3) Randomly initialized learnable prototype alignment → FF	0.211 ± 0.004	0.231 ± 0.001	0.221 ± 0.002
(4) Random markers	0.205 ± 0.001	0.233 ± 0.005	0.219 ± 0.003
(5) 25% markers	0.207 ± 0.003	0.230 ± 0.007	0.219 ± 0.005
(6) 50% markers	0.218 ± 0.0009	0.240 ± 0.005	0.229 ± 0.003
(7) 75% markers	$\mathbf{0.232 \pm 0.001}$	0.244 ± 0.003	0.238 ± 0.002
CellRefine → FF	$\mathbf{0.232 \pm 0.002}$	$\mathbf{0.245 \pm 0.001}$	$\mathbf{0.239 \pm 0.002}$

Table 12: Ablation study isolating the contribution of marker biology for zero-shot gene perturbation response prediction. Results are reported as mean ± standard deviation in Pearson correlation coefficient, where higher is better.

λ_1	Liver	Myeloid
0	0.321 ± 0.012	0.366 ± 0.006
0.1×0.5	0.320 ± 0.009	0.366 ± 0.008
0.3×0.5	0.323 ± 0.014	0.367 ± 0.008
$\mathbf{1 \times 0.5}$	$\mathbf{0.382 \pm 0.008}$	$\mathbf{0.391 \pm 0.006}$
3×0.5	0.362 ± 0.010	0.387 ± 0.006
10×0.5	0.341 ± 0.009	0.362 ± 0.009

Table 13: Sensitivity analysis of λ_1, the prototype loss coefficient. Results are reported as mean ± standard deviation.

In the spatial transcriptomics imputation task presented in Table 11, we observe the same behavior as in the previous task. The compute-matched and content-free controls stay near the baseline, while performance again increases with marker coverage and peaks for *CellRefine* (0.520).

In the zero-shot gene perturbation response prediction task presented in Table 12, the same observation continues. The compute-matched and content-free controls remain at 0.21–0.22, the marker-coverage ladder rises, and *CellRefine* attains 0.239. Although the gains here are small relative to the other two tasks, the low run-to-run variance means the improvement of *CellRefine* over the baseline and the content-free controls is nonetheless distinguishable from noise, and the direction of the trend is consistent with the other two tasks.

D.5 Loss Function Hyperparameter Sensitivity Analysis

For selecting values for $\boldsymbol{\lambda_1}$, $\boldsymbol{\lambda_2}$, and $\boldsymbol{\lambda_3}$, we initialized the loss weights heuristically to balance the magnitudes of the four loss terms, then refined them with a coarse grid search on a held-out validation set, yielding $\boldsymbol{\lambda_1}$ = 0.5, $\boldsymbol{\lambda_2}$ = 1, and $\boldsymbol{\lambda_3}$ = 0.03. We performed a sensitivity analysis on the two most batch-affected datasets, Liver and Myeloid, where the regularization terms are expected to matter most, and sensitivity is therefore most informative. We use a one-at-a-time protocol: each coefficient is varied over a multiplicative grid — 0, 0.1×, 0.3×, 1×, 3×, 10× of its selected value, while the other two are held fixed at their selected values. In every table, the 1× row corresponds to the configuration used throughout the paper and is common to all three tables, while the 0 row removes that loss term (i.e., the corresponding component ablation). All entries are macro-F1 (mean ± s.d. over three runs).

Results of loss parameter sensitivity analysis for cell identity prediction are presented in Table 13, Table 14, and Table 15. Across all three coefficients, the selected value (1×) attains the best macro-F1 on both datasets. The lineage weight $\boldsymbol{\lambda_2}$ is robust: performance is flat to within run-to-run variation across the full 0.1×–10× range, because its margin-based separation loss vanishes once related subtypes are separated and therefore has no effect once the margin is met. For the prototype weight $\boldsymbol{\lambda_1}$, below the selected value the marker signal is too weak to reshape the embedding, and performance falls to the ablation level (Liver ≈ 0.32). The GMVE weight $\boldsymbol{\lambda_3}$ is stable up to the selected value but degrades sharply when over-weighted (Liver 0.382→0.276 at 1×→10×), consistent with a KL term that induces posterior collapse in the post-pretraining stage.

λ_2	Liver	Myeloid
0	0.363 ± 0.011	0.370 ± 0.010
0.1×1	0.371 ± 0.008	0.370 ± 0.009
0.3×1	0.375 ± 0.009	0.373 ± 0.006
$\mathbf{1 \times 1}$	$\mathbf{0.382 \pm 0.008}$	$\mathbf{0.391 \pm 0.006}$
3×1	0.380 ± 0.010	0.390 ± 0.009
10×1	0.381 ± 0.009	0.390 ± 0.011

Table 14: Sensitivity analysis of λ_2, the lineage loss coefficient. Results are reported as mean $\pm$ standard deviation.

λ_3	Liver	Myeloid
0	0.361 ± 0.006	0.372 ± 0.009
0.1×0.03	0.361 ± 0.013	0.372 ± 0.012
0.3×0.03	0.364 ± 0.014	0.373 ± 0.009
$\mathbf{1 \times 0.03}$	$\mathbf{0.382 \pm 0.008}$	$\mathbf{0.391 \pm 0.006}$
3×0.03	0.303 ± 0.010	0.351 ± 0.009
10×0.03	0.276 ± 0.020	0.348 ± 0.009

Table 15: Sensitivity analysis of λ_3, the GMVE loss coefficient. Results are reported as mean $\pm$ standard deviation.

D.6 Compute overhead comparison

CellRefine incurs a one-time cost relative to the initial baseline foundation model, which performs continued MLM/AR post-training in place of *CellRefine* post-pretraining. In Table 16, we report training times for full finetuning Geneformer and Geneformer + *CellRefine*, for the cell identity prediction task (secs) (post-pretraining + fine-tuning) on three datasets. Both methods are trained to convergence under an identical early-stopping criterion. So the reported numbers are each method's genuine time-to-convergence rather than a fixed compute budget.

The overhead is one-time, and to place it in context, it is incurred once per dataset during post-pretraining; it does not recur, and it does not affect deployment. The fine-tuned model has the same architecture and size as the backbone, so *CellRefine* adds nothing to inference latency or memory. Part of this cost is also recovered downstream: by structuring the latent space beforehand, the post-pretraining stage provides a better-initialized embedding, so the subsequent fine-tuning converges faster than fine-tuning from the MLM checkpoint, partially offsetting the added cost.

D.7 Latent space qualitative analysis

Figure 6 illustrates the effect of refinement on the structure of cell embeddings using UMAP visualization. Prior to refinement (Figure 8a), embeddings learned via masked language modeling (MLM) capture coarse-grained structure, but closely related cell types—particularly niche or rare populations such as PP and certain endocrine subtypes—remain poorly separated, with substantial overlap between clusters. After applying *CellRefine* (Figure 8b), the embedding space exhibits markedly improved organization, with clearer cluster boundaries and enhanced separation of fine-grained cell types. Notably, niche cell populations that were previously entangled become more distinctly localized, suggesting that the refinement process improves the model's ability to resolve subtle biological differences and enhances discriminability among closely related cell states.

Figure 7 shows the UMAP visualization of cell embeddings for the liver dataset before and after post-training with *CellRefine*LoRA. Prior to refinement (Figure 7a), embeddings obtained after MLM with LoRA capture coarse cellular structure, but several related cell populations—such as inflammatory and non-inflammatory macrophages, LSEC subtypes (periportal and central vein), and lymphocyte populations—exhibit significant overlap and diffuse boundaries. After applying *CellRefine* LoRA (Figure 7b), the embedding space becomes more structured, with improved separation between biologically distinct cell types. In particular, closely related macrophage subpopulations and endothelial subtypes become more clearly delineated, while lymphocyte clusters (e.g.,

Method	Blood	Liver	Myeloid
Geneformer	18124.0813	5072.1429	15309.3107
Geneformer + *CellRefine*	22039.2802	7200.2811	19322.2311

Table 16: Computational overhead comparison between the initial foundation model and foundation model with *CellRefine* post-pretraining.



(a) UMAP visualization of cell embeddings of the Pancreas dataset embeddings after applying MLM, before applying *CellRefine*.

(b) UMAP visualization of cell embeddings of the Pancreas dataset after post-pretraining with *CellRefine*.

Figure 6: Visualization of latent embeddings for cells in the Pancreas dataset Chen et al. (2023) before (Figure 8a) and after (Figure 8b) post-pretraining with *CellRefine*. Both instances correspond to the alignment stage prior to finetuning. Regions corresponding to cell clusters are outlined with dashed lines.

T/NK cells) are more compact and distinct. These results indicate that lightweight post-training via LoRA is sufficient to enhance the representation quality, enabling better resolution of fine-grained cellular heterogeneity in the liver dataset.

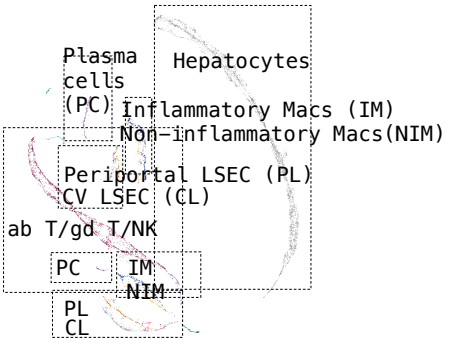
Figure 9 presents the UMAP visualization of cell embeddings for the LivST2 dataset before and after applying *CellRefine*, followed by fine-tuning for cell identity prediction. Prior to refinement (Figure 9a), the embedding space captures broad cellular structure, with major populations such as hepatocytes, macrophages, cholangiocytes, LSECs, and hepatic stellate cells (HSCs) partially organized. However, several of these cell types exhibit overlapping regions and blurred boundaries, particularly among closely related epithelial and endothelial populations. After applying *CellRefine* (Figure 9b), the embeddings become substantially more structured, with clearer separation between distinct cell types. Notably, hepatocytes form a well-defined cluster, while smaller and more specialized populations such as cholangiocytes and LSECs are more distinctly localized. Additionally, macrophages and HSCs become more compact and separable from neighboring clusters. These results demonstrate that CellRefine improves the representation of spatial transcriptomics data, enhancing the model's ability to resolve both major and niche cell populations during downstream cell identity prediction.

E Experimental Settings

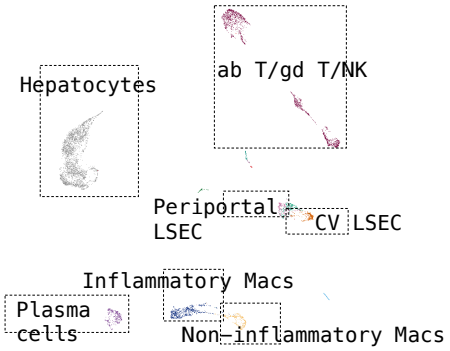
We perform the processing settings below across all datasets and training tasks.

General settings

- Perform quality control on all datasets, removing unrecognized cell types such as 'other', 'unknown' as well as cells with too few captured genes.
- For all tasks, we perform three random iterations and report the mean.

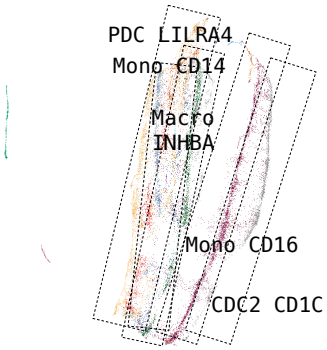


(a) UMAP visualization of cell clusters of liver dataset after applying MLM LoRA.



(b) UMAP visualization of cells in the liver dataset after post-pretraining with *CellRefine* LoRA.

Figure 7: Visualization of latent embeddings for cells in the liver dataset Lin et al. (2020) before (Figure 7a) and after (Figure 7b) post-pretraining with *CellRefine* LoRA. Both instances correspond to the alignment stage prior to finetuning. Regions corresponding to cell clusters are outlined with dashed lines.



(a) UMAP visualization of cell embeddings of the Myeloid dataset embeddings after applying MLM, before applying *CellRefine*.

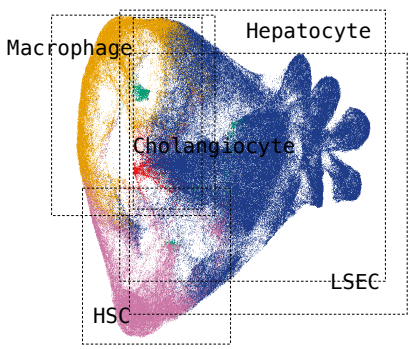


(b) UMAP visualization of cell embeddings of the Myeloid dataset after post-pretraining with *CellRefine*.

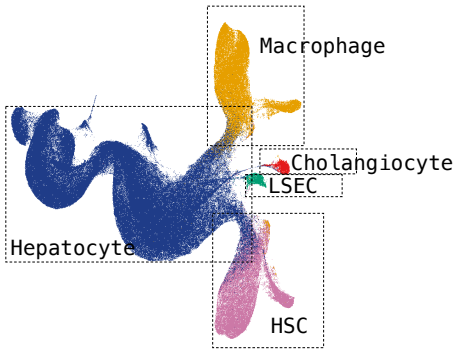
Figure 8: Visualization of latent embeddings for cells in the Myeloid dataset before (Figure 8a) and after (Figure 8b) post-pretraining with *CellRefine*. Both instances correspond to the alignment stage prior to finetuning. Regions corresponding to cell clusters are outlined with dashed lines.

- All tasks were trained with early stopping based on validation loss, with a patience of 3 epochs.
- For MLM tasks, all models are trained until MLM loss reaches 3, following Geneformer Theodoris et al. (2023) pretraining.
- Blood cell dataset Zheng et al. (2017) uses a 80:20 train:test split and all other datasets use separate datasets for training and testing.

Training All training was conducted in PyTorch v2.6.0 (CUDA 12.4 support), leveraging the Hugging Face Transformers version 4.44.2. We optimized model parameters using AdamW with a linear warmup phase of 1,000 steps to a peak learning rate of 1e-5, after which the learning rate was linearly decayed. Weight decay was set to 0.001. We ran the experiments on AMD EPYC 7702 64-core



(a) UMAP visualization of cell embeddings of the LivST2 dataset embeddings after aligning with MLM and finetuning for cell identity prediction task.



(b) UMAP visualization of cell embeddings of the LivST2 dataset after post-pretraining with *CellRefine* and finetuning for cell itentity prediction task.

Figure 9: Visualization of latent embeddings for tail cells in the LivST2 dataset before (Figure 9a) and after (Figure 9b) applying *CellRefine*, and after finetuning for cell identity prediction task.

	Hyperparameter	**Value**
Cell Encoder (f)	Vocab size	20275
	Hidden size	768
	Number of hidden layers	12
	Max sequence length	4096
	Number of attention heads	12
	Dropout	0.02
	Activation	ReLU
	LayerNorm eps	1e-12
Marker Encoder (f_{Marker})	Vocab size	20275
	Hidden size	768
	Number of hidden layers	2
	Max sequence length	200
	Number of attention heads	12
	Dropout	0.01
	Activation	ReLU
	LayerNorm eps	1e-12
LoRA Configuration	Rank (r)	8
	Alpha (α)	16
	Dropout	0.05
	Target modules	query, key, value (attention layers)
	Bias	none
	Task type	encoder-only
Post-pretraining	Optimizer	AdamW
	Scheduler	Linear
	Maximum learning rate	1e-5
	Warmup steps	1,000
	Weight decay	1e-3
	Batch size	12

Table 17: Experiment configurations

CPUs with 4 NVIDIA A100 GPUs with 80GB VRAM. Hyperparameter settings for the cell encoder, marker encoder, LoRA adapters, and post-pretraining are included in the Table 17

F DATASET DETAILS

Blood The Blood dataset Zheng et al. (2017) is a particularly challenging benchmark, comprising 68,450 peripheral blood mononuclear cells (PBMCs). It includes 11 closely related cell subtypes with subtle transcriptional differences, making accurate discrimination difficult. Additionally, the dataset exhibits a strongly imbalanced cell-type distribution, where certain populations are under-represented. We use 80:20 train:test split of dataset for the cell identification task.

Pancreas The Pancreas dataset contains scRNA-seq data from five independent studies of human pancreas cells, reprocessed by Chen et al. (2023). Following scGPT Cui et al. (2024), we split the five datasets to two reference datasets and three query datasets. The reference set contains 10,600 cells of 13 cell types, and the query set contains 4218 cells of 11 cell types. We used this dataset for the cell identification task.

Liver The Liver dataset sourced from Lin et al. (2020) is a combination of the macParland and the Aizarani datasets. We utilized the macParland dataset, comprising 14 cell types and 8444 cells, and the reference set, and the Aizarani dataset, which includes 9162 cells and seven cell types that are present in the macParland dataset as the query dataset. We used this dataset for the cell identification task.

LiverST1 LiverST1 is a human liver spatial transcriptomics dataset sourced from 10x Genomics 10X Genomics (2023). It contains 90,408 cells and five cell types. This dataset was used for cell identification and spatial transcriptomics imputation tasks. This dataset was used to report performance when the models were trained with LiverST2 dataset.

LiverST2 LiverST2 is a human liver spatial transcriptomics dataset sourced from 10x Genomics 10X Genomics (2023), containing 309,808 cells and five same cell types included in LivST1 dataset. We used this dataset for cell identity prediction and spatial transcriptomics imputation tasks, and was used to report performance when the models were trained with LiverST1 dataset.

Myeloid Myeloid dataset Cheng et al. (2021) can be accessed from Gene Expression Omnibus (GEO) database using accession number GSE154763. While the dataset contains nine different cancer types, following Cui et al. (2024), we used six cancer types: UCEC, PAAD, THCA, LYM, cDC2, and kidney in the train set and three cancer types: MYE, OV-FTC, and ESCA in the test set.

MS The MS dataset Schirmer et al. (2019) has nine healthy samples, and 12 MS samples are included in the dataset. Following scGPT Cui et al. (2024), we use the control samples as the reference set and MS samples as the query set, creating a setting to test on out-of-distribution data. The reference set has a cell count of 7,844, and the query set has a cell count of 13,468.

Lung The lung dataset Kim et al. (2020) consists of 14 lung adenocarcinoma samples, with 30,472 cells. The dataset is publicly available at (https://www.weizmann.ac.il/sites/3CA/lung). We partitioned it into a reference set of ten samples and a query set of four samples. The resulting reference set contains 24,746 cells, and the query set contains 7,747 cells.

G EVALUATION METRICS

G.1 CELL TYPE ANNOTATION

To evaluate performance on the cell type annotation task, we consider a multi-class classification setting with K cell types. Given that several datasets are highly imbalanced, we primarily report the **macro F1-score**, which assigns equal importance to each class regardless of its frequency.

Let y_i and $\hat{y}_i$ denote the ground-truth and predicted labels for sample i, respectively. For each class $k \in \{1, \ldots, K\}$, we define true positives (TP_k), false positives (FP_k), and false negatives (FN_k).

We first compute class-wise precision and recall:

$$\text{Precision}_k = \frac{TP_k}{TP_k + FP_k}, \quad \text{Recall}_k = \frac{TP_k}{TP_k + FN_k} \tag{16}$$

Using these, the F1-score for each class is:

$$F1_k = \frac{2 \cdot \text{Precision}_k \cdot \text{Recall}_k}{\text{Precision}_k + \text{Recall}_k} \tag{17}$$

The macro F1-score is defined as the unweighted average across all classes:

$$\text{Macro-F1} = \frac{1}{K} \sum_{k=1}^{K} F1_k \tag{18}$$

For completeness, we also report overall accuracy:

$$\text{Accuracy} = \frac{\sum_{k=1}^{K} TP_k}{N} \tag{19}$$

where N is the total number of samples.

In addition, we compute the weighted F1-score, which accounts for class imbalance by weighting each class-specific F1-score by its support n_k (the number of samples in class k):

$$\text{Weighted-F1} = \frac{1}{N} \sum_{k=1}^{K} n_k \cdot F1_k \tag{20}$$

While accuracy and weighted F1-score reflect overall performance dominated by frequent classes, the macro F1-score provides a balanced evaluation across all cell types and is therefore our primary metric.

G.2 Spatial Transcriptomics Imputation

To evaluate performance on spatial transcriptomics imputation, we assess the agreement between imputed gene expression profiles and ground-truth measurements. Let $\mathbf{x}_i \in \mathbb{R}^G$ denote the ground-truth gene expression vector and $\hat{\mathbf{x}}_i \in \mathbb{R}^G$ the imputed vector for spot (or cell) i, where G is the number of genes and $i \in \{1, \ldots, N\}$ indexes spatial locations.

We employ two complementary metrics: **Pearson correlation** and **cosine similarity**, both computed at the spot level and averaged across all samples.

Pearson Correlation. Pearson correlation measures the linear relationship between imputed and ground-truth expression values. For each sample i, it is defined as:

$$r_i = \frac{\sum_{g=1}^{G} (x_{ig} - \bar{x}_i)\left(\hat{x}_{ig} - \bar{\hat{x}}_i\right)}{\sqrt{\sum_{g=1}^{G} (x_{ig} - \bar{x}_i)^2} \sqrt{\sum_{g=1}^{G} \left(\hat{x}_{ig} - \bar{\hat{x}}_i\right)^2}} \tag{21}$$

where $\bar{x}_i = \frac{1}{G} \sum_{g=1}^{G} x_{ig}$ and $\bar{\hat{x}}_i = \frac{1}{G} \sum_{g=1}^{G} \hat{x}_{ig}$ denote the mean expression values across genes for the ground-truth and imputed vectors, respectively.

The final Pearson correlation score is obtained by averaging across all samples:

$$\text{Pearson} = \frac{1}{N} \sum_{i=1}^{N} r_i \tag{22}$$

Cosine Similarity. Cosine similarity measures the similarity in direction between imputed and ground-truth expression vectors, independent of their magnitudes. For each sample i, it is defined as:

$$\text{Cosine}_i = \frac{\sum_{g=1}^{G} x_{ig}\hat{x}_{ig}}{\sqrt{\sum_{g=1}^{G} x_{ig}^2}\sqrt{\sum_{g=1}^{G} \hat{x}_{ig}^2}} \tag{23}$$

The overall cosine similarity is computed as:

$$\text{Cosine} = \frac{1}{N}\sum_{i=1}^{N} \text{Cosine}_i \tag{24}$$

Pearson correlation captures linear agreement between imputed and true expression levels, while cosine similarity evaluates the consistency of expression patterns irrespective of scale. Together, these metrics provide a comprehensive assessment of imputation quality.

G.3 Gene Perturbation Response Prediction

To evaluate performance on gene perturbation response prediction, we measure the agreement between predicted and ground-truth *differential gene expression* (DGE) profiles. Differential gene expression quantifies the change in gene expression induced by a perturbation. For each gene g under perturbation i, it is defined as the difference between post-perturbation and pre-perturbation expression levels:

$$\Delta x_{ig} = x_{ig}^{\text{post}} - x_{ig}^{\text{pre}} \tag{25}$$

where x_{ig}^{pre} and x_{ig}^{post} denote the expression of gene g before and after the perturbation, respectively.

Let $\Delta\mathbf{x}_i \in \mathbb{R}^G$ denote the true differential expression vector and $\widehat{\Delta\mathbf{x}}_i \in \mathbb{R}^G$ the predicted differential expression vector for perturbation i, where G is the number of genes and $i \in \{1, \ldots, N\}$ indexes perturbation conditions.

We use the **Pearson correlation coefficient** to quantify similarity between predicted and true DGE profiles. For each perturbation i, the Pearson correlation is computed as:

$$r_i = \frac{\sum_{g=1}^{G}\left(\Delta x_{ig} - \overline{\Delta x}_i\right)\left(\widehat{\Delta x}_{ig} - \overline{\widehat{\Delta x}}_i\right)}{\sqrt{\sum_{g=1}^{G}\left(\Delta x_{ig} - \overline{\Delta x}_i\right)^2}\sqrt{\sum_{g=1}^{G}\left(\widehat{\Delta x}_{ig} - \overline{\widehat{\Delta x}}_i\right)^2}} \tag{26}$$

where $\overline{\Delta x}_i = \frac{1}{G}\sum_{g=1}^{G}\Delta x_{ig}$ and $\overline{\widehat{\Delta x}}_i = \frac{1}{G}\sum_{g=1}^{G}\widehat{\Delta x}_{ig}$ denote the mean differential expression across genes for the ground-truth and predicted profiles, respectively.

The final evaluation metric is obtained by averaging across all perturbations:

$$\text{Pearson} = \frac{1}{N}\sum_{i=1}^{N} r_i \tag{27}$$

Pearson correlation captures the linear agreement between predicted and true differential expression patterns across genes, providing a robust measure of how well the model recovers the direction and relative magnitude of gene-level responses to perturbations.

H Cell Ontologies and Type Distributions

Figure 10 illustrates the long-tail distributions of cell types in many of the single-cell datasets.

I Long tail quantification

To quantify the degree of class imbalance across datasets, we estimated *power-law tail exponents* (α) from the distribution of cell type frequencies. Specifically, we fit a log–log regression to the

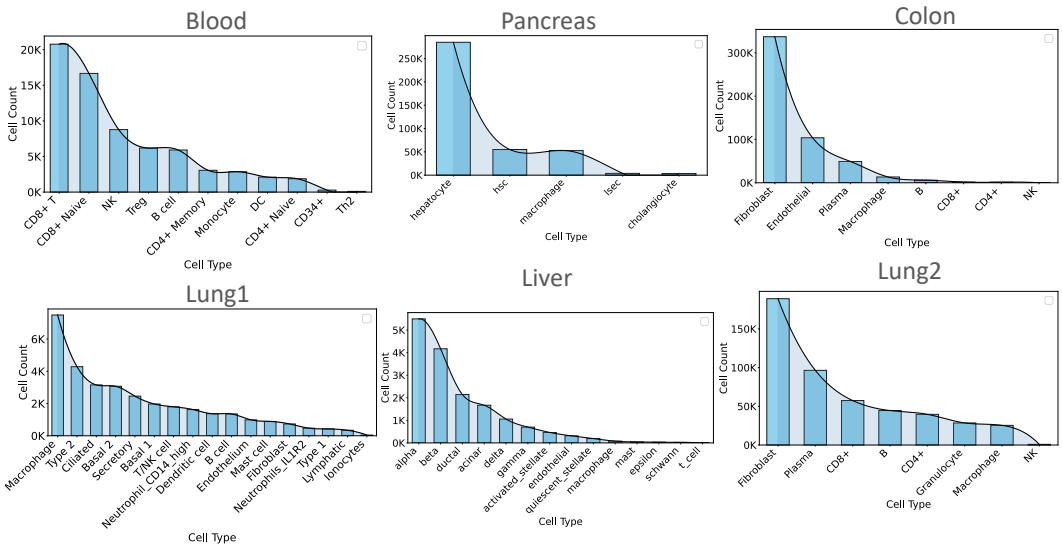


Figure 10: Long-tailed distributions of single-cell datasets

complementary cumulative distribution function (CCDF) of the rarest 30% of categories, following common practice in heavy-tailed data analysis. The CCDF of a power-law distributed variable X is given by

$$P(X \geq x) \sim x^{-\alpha}$$

where smaller values of α correspond to heavier tails, reflecting stronger dominance of a few frequent types and a larger fraction of rare ones. As summarized in Table 18, the fitted exponents range from $\alpha \approx 0.2$ (Custom3) to $\alpha \approx 0.6$ (MS), confirming that single-cell datasets exhibit highly skewed and heavy-tailed distributions. For the Liver dataset, the number of categories ($n = 5$) was too small to reliably estimate a tail exponent. These results provide a quantitative basis for treating single-cell learning tasks as inherently long-tailed recognition problems, motivating the need for methods that explicitly address rare classes.

Dataset	**# Cell Types**	**Estimated Tail Exponent** (α)	**Fit** R^2
Pancreas	17	0.45	0.583
MS	14	0.631	0.693
Blood	11	0.37	1.000

Table 18: Estimated tail exponents for cell type distributions across datasets. Smaller α values indicate heavier tails.

J ADDITIONAL DATASET DETAILS

Dataset	Number of sources	Downstream task	Cell type	Count
LiverST1	1	Cell identity prediction Spatial transcriptomics imputation	hepatocyte	228,654
			hsc	41,043
			macrophage	34,861
			cholangiocyte	3,110
			lsec	2,140
LiverST2	1	Cell identity prediction Spatial transcriptomics imputation	hepatocyte	56,286
			macrophage	17,808
			hsc	13,992
			lsec	1,690
			cholangiocyte	632
Blood	1	Cell identity prediction	CD8+ Cytotoxic T	20,757
			CD8+/CD45RA+ Naive Cytotoxic	16,645
			CD56+ NK	8,775
			CD4+/CD25 T Reg	6,185
			CD19+ B	5,877
			CD4+/CD45RO+ Memory	3,059
			CD14+ Monocyte	2,847
			Dendritic	2,095
			CD4+/CD45RA+/CD25$^-$ Naive T	1,871
			CD34+	242
			CD4+ T Helper2	97
Pancreas	5	Cell identity prediction	alpha	5,147
			beta	3,972
			ductal	1,704
			acinar	1,353
			delta	981
			PP	638
			PSC	597
			endothelial	289
			macrophage	52
			mast	32
			epsilon	28
			schwann	13
			t_cell	7
			MHC class II	5

Table 19: Dataset composition and downstream tasks across the evaluated benchmarks (1/2).

Dataset	Number of sources	Downstream task	Cell type	Count
Liver	2	Cell identity prediction	Hepatocytes	6,587
			NK, NKT and T cells	5,084
			LSEC	1,994
			macrophage	1,192
			cholangiocytes	1,141
			MSEC	566
			Plasma cells	511
			B cells	258
			Erythroid cells	93
			Stellate and myofibroblasts	65
Myeloid	9	Cell identity prediction	Mono CD16	11,214
			Mono CD14	10,546
			Macro C1QC	7,927
			cDC2 CD1C	7,883
			Macro NLRP3	3,903
			Mast KIT	3,868
			Macro LYVE1	3,419
			Macro SPP1	3,279
			Macro IL1B	2,363
			Macro GPNMB	2,264
			Macro INHBA	2,047
			cDC2 CXCR4hi	1,851
			Mono CD14CD16	1,608
			Macro FN1	1,319
			cDC2 CD1A	1,209
			cDC2 IL1B	1,208
			pDC LILRA4	786
			cDC3 LAMP3	740
			Macro ISG15	726
			cDC2 FCN1	720
			cDC1 CLEC9A	673
			cDC2 ISG15	302
			cDC2 CXCL9	171

Table 20: Dataset composition and downstream tasks across the evaluated benchmarks (2/2).