Foundation Models for Analyzing Single-Cell RNA Sequence data

Amirreza Naziri

A Thesis submitted to the Faculty of Graduate Studies in Partial Fulfillment of the Requirements for the Degree of Master of Science

Graduate Program in Electrical and Computer Science
York University
Toronto, Ontario

December 2025

Abstract

Single-cell RNA sequencing (scRNA-seq) measures gene expression in individual cells, offering deep insight into cellular heterogeneity, development, and disease. Transformer-based foundation models have become central to single-cell RNA-sequencing analysis, yet most rely on uniform random masking during pretraining, a strategy misaligned with the sparsity, heterogeneity, and zero inflation characteristic of scRNA-seq data. To assess how these models behave under realistic biological variation, we first perform a comprehensive evaluation of four widely used single-cell foundation models (Geneformer, scBERT, scFoundation, and scGPT) across three diverse datasets. This benchmarking reveals substantial variability in model performance, including systematic weaknesses on rare cell populations and degraded accuracy in clinically challenging conditions. Motivated by the broader limitations of random masking in Foundation models, we introduce Multinomial Attention Masking (MAM), a biologically informed masking strategy that leverages trainable latent representations and cross-attention to identify informative gene positions during pretraining. Across all datasets, models pretrained with MAM consistently achieve higher downstream cell-type classification accuracy than those trained with uniform masking and, in several cases, outperform the original pretrained backbones. Biological validation further demonstrates that MAM preferentially selects highly expressed and functionally meaningful genes, indicating that its improvements stem from capturing biologically relevant structure rather than from increased algorithmic complexity. This work improves the reliability and utility of single-cell foundation models for researchers and clinicians alike.

Originality Statement

This dissertation is based on the author's published and submitted research articles. The core ideas, methods, and experimental results presented throughout the thesis were developed from the following papers:

- **Published (AAAI Symposium on Drug Discovery, 2025)**
  A. Naziri, A. Asgari, A. An, E. Sachlos, and L. Seyyed-Kalantari, *"From Bias to Breakdown: Benchmarking Failure Mode Analysis of Single-cell RNA Sequencing Foundation Models in Acute Myeloid Leukemia."*

- **Published (NeurIPS AI4D3 Workshop, 2025)**
  A. Naziri, A. Asgari, E. Sachlos, A. An, and L. Seyyed-Kalantari, *"Improving Classification of Cell Types in Acute Myeloid Leukemia with Self-guided Masking Technique."*

- **Under Submission**
  A. Naziri, A. Asgari, E. Sachlos, A. An, and L. Seyyed-Kalantari, *"Multinomial Attention Masking for Robust Pretraining of Single-cell Foundation Models."*

The thesis integrates, extends, and unifies the contributions from these works. Material from the published papers has been adapted with permission where required. All experimental implementations, analyses, and figures appearing in the dissertation were produced by the author unless otherwise stated.

Acknowledgements

I would like to express my deepest gratitude to my supervisor, Prof. Laleh Seyyed-Kalantari, for her guidance, encouragement, and continuous support throughout this research. Her mentorship has shaped every stage of this work and has greatly influenced my development as a researcher. I would also like to sincerely thank the committee members, Prof. Aijun An, Prof. Terry Sachlos and Prof. Christo El Morr, for their valuable time, insightful feedback, and constructive suggestions that strengthened this work.

I am sincerely thankful to York University for providing a supportive and inspiring academic environment. I especially acknowledge the Center for AI & Society (AI-EGS) for fostering interdisciplinary research and offering valuable resources that contributed to this project. I also extend my appreciation to the Vector Institute for providing access to high-performance computing resources that made large-scale experimentation possible.

This work was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) through a Discovery Grant, as well as the Connected Minds initiative, funded by the Canada First Research Excellence Fund (CFREF). Their support played a crucial role in enabling the computational and methodological exploration carried out in this study.

Finally, I am grateful to my colleagues, collaborators, and friends especially Arash Asgari for their helpful discussions, feedback, and ongoing encouragement.

Contents

Page

Abstract **ii**

Originality Statement **iii**

Acknowledgements **iv**

Contents **v**

List of Tables **vii**

List of Figures **ix**

1 Introduction **1**

1.1 Background: Single-cell RNA Sequencing and Its Impact 1

1.2 Challenges in Analyzing Large-scale scRNA-seq Data 1

1.3 Emergence of Machine Learning in scRNA-seq Analysis 1

1.4 Transformer and Foundation Model Revolution 2

1.4.1 Transformer Architecture Overview 2

1.4.2 Pre-training and Fine-tuning Paradigm in Transformers 5

1.5 Foundation Models in Biology and Genomics 6

1.6 Limitations of Current Foundation Models in scRNA-seq 7

1.7 Research Gap and Motivation 9

1.8 Proposed Approach 10

2 Literature Review **11**

2.1 Classical scRNA-seq Analysis Methods 11

2.2 Deep Learning Methods for scRNA-seq Analysis 12

2.3 scRNA-seq Foundation Models and Transformers 13

2.4 Pre-training and Fine-tuning Paradigms 15

2.5 Few-Shot and Zero-Shot Learning 16

2.6 Attention-Based Masking and Adaptive Strategies 17

2.7 Challenges in scRNA-seq Data 18

3 Method **20**

3.1 Dataset 20

3.1.1 Description of Individual Datasets 20

3.1.2 Data Preprocessing 23

3.2 Foundation Models 24

3.3 Foundation Models Evaluation 25

3.3.1 Fine-Tuning Pipeline . . . 25
3.3.2 Dataset-Specific Evaluation . . . 26
3.3.3 Summary . . . 27
3.4 Multinomial Attention Masking (MAM) Algorithm . . . 28
3.4.1 Backbone Foundation Model . . . 28
3.4.2 Latent Cross-Attention Module . . . 28
3.4.3 Multinomial Attention Masking (MAM) . . . 29
3.4.4 Cross-Attention Context Injection Strategies . . . 30
3.4.5 Self-supervised Pretraining . . . 31
3.4.6 Supervised Finetuning Setup . . . 32
3.4.7 Biological Evaluation . . . 34
3.4.8 Summary . . . 34

4 Experiments **35**
4.1 Foundation Model Evaluation . . . 35
4.1.1 Training Hyper-parameters . . . 35
4.1.2 Computing Resources . . . 35
4.1.3 Overall Model Performance Across Datasets . . . 36
4.1.4 In-Depth Analysis on the Hematopoietic Niche Dataset . . . 38
4.1.5 Summary . . . 40
4.2 Multinomial Attention Masking (MAM) Algorithm . . . 42
4.2.1 Training Hyper-parameters . . . 42
4.2.2 Computing Resources . . . 44
4.2.3 Experiment Results . . . 44
4.2.4 Biological Validation Results . . . 49
4.2.5 Summary . . . 51

5 Conclusion **53**
5.1 Summary of Findings . . . 53
5.2 Contributions and Impact . . . 54
5.3 Limitations . . . 54
5.4 Future Directions . . . 55

Bibliography **57**

Appendices **74**

A Partial Finetuning Results (MAM) **74**

B Linear Probe Results (MAM) **75**

List of Tables

1 Summary of foundation models evaluated in this study. 25
2 Training hyper-parameters used for fine-tuning each foundation model. 35
3 Compute resources and end-to-end runtime for all foundation model evaluations. Each experiment used 4 × NVIDIA A40 GPUs (48 GB each). Runtime estimates reflect fine-tuning and evaluation only. 36
4 F1-score performance of four foundation models across three datasets. Results are reported as the mean ± confidence interval from 5-fold cross-validation on the test set, with each fold held out once. 38
5 Hyperparameters for self-supervised pretraining (MAM). 43
6 Hyperparameters for supervised finetuning (MAM). 43
7 Compute resources and end-to-end runtime for every experiment group (MAM). Each experiment used 4 × NVIDIA A40 GPUs (48 GB each). 44
8 Partial-finetune cell-type classification, comparing MAM (Ours) F1-score vs random masking. (FT: finetune, P: pretraining, Orig. = original backbone weights [1], all tokens: masked all attention-suggested tokens; 80/10/10: the standard random 80/10/10 scheme, Injection: which stream(s) received cross-attention output). Bolded numbers are the best. Results are reported as the mean ± confidence interval from 5-fold cross-validation on the test set, with each fold held out once. 46
9 Linear-probe cell-type classification (F1-score). Same notation as Table 8. 47
10 Appendix: Partial-finetune results on the Hematopoietic Niche dataset. (FT = finetune, P = pretraining, Orig. = original backbone weights, all tokens = masked all attention-suggested tokens, 80/10/10 = the standard random 80/10/10 scheme, Injection: which stream(s) received cross-attention output). Here Orig. comes from [1]. Results are reported as the mean ± confidence interval from 5-fold cross-validation on the test set, with each fold held out once. 74
11 Appendix: Partial-finetune results on the PBMC68K dataset. Same notation as Table 10. Here Orig. comes from [1]. Results are reported as the mean ± confidence interval from 5-fold cross-validation on the test set, with each fold held out once. 74
12 Appendix: Partial-finetune results on the Heart dataset. Same notation as Table 10. Here Orig. comes from [1]. Results are reported as the mean ± confidence interval from 5-fold cross-validation on the test set, with each fold held out once. 75
13 Appendix: Linear-probe results on the Hematopoietic Niche dataset. Same notation as Table 10. Here Orig. comes from [1]. Results are reported as the mean ± confidence interval from 5-fold cross-validation on the test set, with each fold held out once. 75
14 Appendix: Linear-probe results on the PBMC68K dataset. Same notation as Table 10. Here Orig. comes from [1]. Results are reported as the mean ± confidence interval from 5-fold cross-validation on the test set, with each fold held out once. 76

15 Appendix: Linear-probe results on the Heart dataset. Same notation as Table 10. Here Orig. comes from [1]. Results are reported as the mean ± confidence interval from 5-fold cross-validation on the test set, with each fold held out once. 76

List of Figures

1 Overview of the Transformer architecture showing the encoder (left) and decoder (right). Adapted from "Attention Is All You Need" paper [2]. 3

2 UMAP visualization of Hematopoietic Niche scRNA-seq (Acute Myeloid Leukemia) [3]. Each point represents a single cell colored by (top) cell type and (bottom) clinical timepoint. 21

3 UMAP visualization of the PBMC68K dataset, where each point represents a single cell colored by its cell type. 22

4 UMAP visualization of the heart dataset, where each point represents a single cell colored by its cell type. 23

5 Proposed attention-driven masking and context-injection workflow. Learned latent vectors attend to token embeddings via cross-attention (Q, K, V) to produce attention weights A and pooled context O. A multinomial sampling of A yields the mask positions. The masked sequence is passed through encoder and decoder layers, with O injected according to four strategies: (A) encoder-only, (B) decoder-only, (C) both with a shared adapter, or (D) both with separate adapters. 29

6 Supervised finetuning setup for downstream classification. During finetuning, latent tokens, cross-attention layers, and all adapters remain frozen. The MLP outputs a probability distribution over cell types for evaluation using the F1-score. 32

7 Performance of four foundation models (Geneformer, scBERT, scFoundation, scGPT) on Hematopoietic Niche dataset. 37

8 Performance of four foundation models (Geneformer, scBERT, scFoundation, scGPT) on PBMC 68K dataset. 37

9 Performance of four foundation models (Geneformer, scBERT, scFoundation, scGPT) on Heart dataset. 38

10 Performance of four foundation models (Geneformer, scBERT, scFoundation, scGPT) stratified by clinical timepoint. While models perform well on Healthy samples, performance drops in AML, with the largest decline observed in relapse cases. This pattern highlights a systematic gap: foundation models pretrained on mostly healthy single-cell data generalize poorly to clinically challenging disease states, even though our evaluation dataset was evenly balanced across all timepoints. 39

11 Heatmap of F1 scores by cell type across four foundation models. Performance is uneven: common and well-represented populations such as B cells show strong accuracy, while others like CLP, Pre.B exhibit markedly lower scores. In addition, some cell types are totally mistaken by foundation models, resulting in zero in their F1 Score. 40

12 Scatter plot of each gene's masking frequency against its mean expression when masked (experiment 8, Appendix Table 11), illustrating that MAM preferentially targets genes with higher expression levels. 50

13 Top enriched GO biological processes among the most frequently masked genes in the PBMC68K dataset. The x-axis demonstrates the significance of each biological process (y-axis) derived from masked genes. 51

1 Introduction

1.1 Background: Single-cell RNA Sequencing and Its Impact

Single-cell RNA sequencing (scRNA-seq) has significantly advanced our understanding of cellular heterogeneity by allowing the measurement of transcriptomes at the level of individual cells [4, 5]. This powerful technology has led to major discoveries in many areas of cell biology, such as identifying and describing new cell types and rare subpopulations, uncovering complex gene regulatory networks specific to certain cell types, and generating detailed profiles of how individual cells respond to drugs or genetic changes [6, 7, 8].

The analysis of scRNA-seq data generally follows a well-defined computational workflow. This includes key steps such as performing quality control to remove low-quality cells and technical artifacts, applying normalization methods to correct for sequencing depth and technical variation, selecting highly variable genes, reducing dimensionality using techniques like principal component analysis and manifold learning, clustering cells to identify distinct populations, and finally, annotating cell types [9, 10, 11]. These standard approaches, mostly implemented in tools such as Seurat and Scanpy, form the basis for modern scRNA-seq data analysis [12, 13, 14, 15, 16, 17, 18, 19, 20].

1.2 Challenges in Analyzing Large-scale scRNA-seq Data

Despite their wide use and proven effectiveness, classical methods face serious challenges when dealing with the growing size and complexity of modern scRNA-seq datasets [18]. The rapid increase in data generation, with studies now often analyzing millions of cells from hundreds of experimental batches, creates major computational difficulties in terms of memory use, processing time, and algorithm scalability [20]. Traditional methods also struggle to accurately represent the unique features of scRNA-seq data, such as the high number of zeros caused by technical dropouts, the strong sparsity that hides biological signals, and complex batch effects resulting from differences in experimental platforms and conditions [17, 18, 19, 20].

Moreover, common dimensionality reduction methods like principal component analysis rely on linear assumptions that cannot capture the nonlinear gene interactions and hierarchical regulatory patterns typical of biological systems. These limitations also affect the detection of rare cell types, which are biologically important but small in number, and are often missed or misclassified by classical clustering algorithms [21, 22, 23]. Another persistent issue is batch effect correction, as standard normalization techniques may either fail to fully remove technical variation or overcorrect, which can remove real biological signals [24, 25, 26, 27].

1.3 Emergence of Machine Learning in scRNA-seq Analysis

To overcome these limitations, there has been a major shift toward machine learning-based methods. These approaches use deep neural networks to find complex patterns in high-dimensional transcriptomic data and to support more automated and scalable analysis workflows [28]. Deep learning models, especially those based on variational autoencoders, show strong ability in modeling the zero-inflated negative binomial distribution

that characterizes scRNA-seq count data. This allows a more accurate representation of the gene expression process [29].

1.4 Transformer and Foundation Model Revolution

Transformer-based architectures and attention mechanisms, which were first developed in natural language processing, have also been successfully applied to scRNA-seq analysis. They allow models to learn contextual gene representations and to capture long-range relationships between genes [30]. The development of large-scale foundation models (FMs) that are pre-trained on tens of millions of cells from different tissues, conditions, and species represents a major advance [30, 31].

These models learn general patterns of cellular states and gene regulation that can be fine-tuned for specific downstream tasks using only a small amount of labeled data. Self-supervised and contrastive learning methods have also shown great potential, as they take advantage of the natural structure of unlabeled scRNA-seq data to learn meaningful representations that group functionally similar cells together [32].

At the same time, transformer-based foundation models have become an important new direction in artificial intelligence. These models are large neural network architectures that are pre-trained in an unsupervised or self-supervised way on massive collections of unlabeled data, and then fine-tuned for different downstream tasks in many fields [33, 34, 35, 36, 37, 38, 39].

Foundation models were first developed for natural language processing. Key examples include BERT (Bidirectional Encoder Representations from Transformers) [34, 40, 41, 42], which introduced masked language modeling as a pre-training task, and the GPT (Generative Pre-trained Transformer) family [34, 43, 36, 39, 44]. GPT-3, with its 175 billion parameters, showed remarkable few-shot learning ability through autoregressive next-token prediction [34, 36, 44, 45, 46, 47].

The transformer framework later expanded into computer vision through the Vision Transformer (ViT) [48, 49, 50, 51], which redefined image analysis by viewing images as sequences of patches. Masked Autoencoders (MAE) [52, 53, 54, 55] further extended this concept by applying the masking idea from natural language processing to visual learning, reconstructing images from randomly masked patches [52, 56].

1.4.1 Transformer Architecture Overview

The first key component includes the token embedding and positional encoding layers. These layers convert discrete input elements such as words in text, image patches in vision tasks, or gene expression values in biological data into continuous high-dimensional vector representations that the network can process [57, 58, 59, 60, 61, 62, 49, 51]. Because the self-attention mechanism treats input as an unordered set, positional encodings are necessary to provide information about the order of tokens in a sequence [57, 58, 59, 61].

The original transformer uses sinusoidal positional encodings based on sine and cosine functions with different frequencies. These encodings produce unique position-based patterns that allow the model to learn

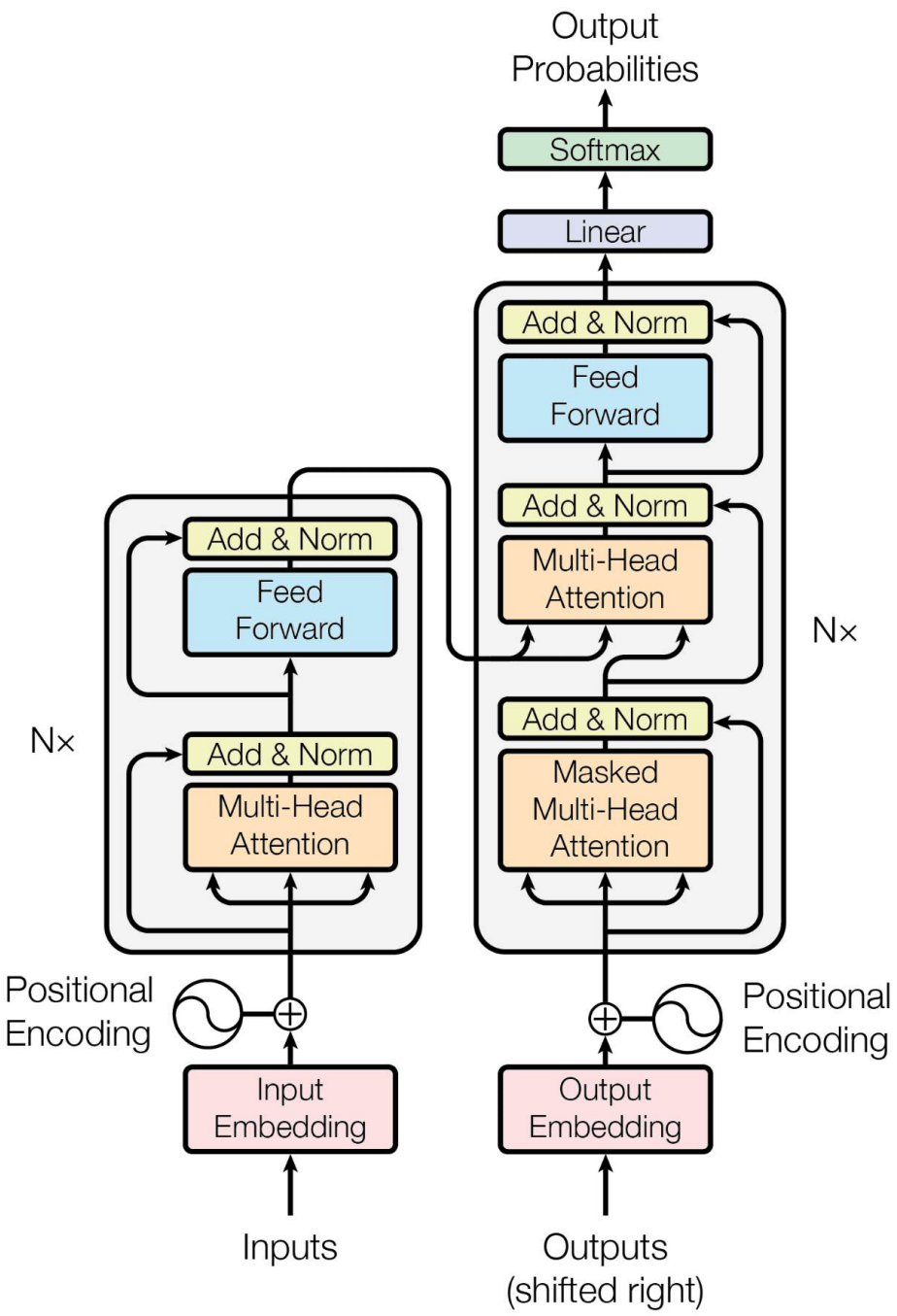

Figure 1: Overview of the Transformer architecture showing the encoder (left) and decoder (right). Adapted from "Attention Is All You Need" paper [2].

relative relationships between tokens [57, 59, 61]. More recent designs use alternatives such as learned positional embeddings, relative positional encodings like RoPE (Rotary Position Embedding), and ALiBi

(Attention with Linear Biases).

Each of these methods offers specific advantages for modeling long-range dependencies and extending model performance beyond the training sequence length [58, 60]. Token embeddings are usually initialized randomly or learned during training, although some architectures use pre-trained embeddings such as word2vec or gene2vec to provide meaningful starting points [41, 62, 63].

The encoder layers make up the second main module of the transformer architecture. They are responsible for combining and transforming embedded representations into high-level contextual encodings through stacked blocks that include multi-head self-attention and position-wise feed-forward networks [35, 41, 57, 64, 65, 66, 62].

The self-attention mechanism allows each position in a sequence to attend to all other positions. It computes attention weights that represent how relevant or important each token is to every other token in the context of the sequence [35, 41, 67, 64, 68, 62]. This process uses three learned linear projections for each token: Query (Q), Key (K), and Value (V) vectors. Attention scores are obtained by taking scaled dot products between the query and key vectors, and then applying a softmax function to produce attention weights. These weights are then used to compute weighted sums of the value vectors [41, 67, 64, 62].

Multi-head attention extends this idea by performing several attention computations in parallel, each with different learned linear transformations. This enables the model to attend to different types of information from multiple representation subspaces at once [40, 41, 64, 62]. This ability to process sequences in parallel is a major advantage over recurrent neural networks (RNNs), which handle data sequentially. As a result, transformers can train much faster and scale to very large model sizes [33, 35, 36, 37, 36, 69, 70].

Each encoder block typically follows a consistent structure: a multi-head self-attention layer followed by layer normalization and a residual connection, then a position-wise feed-forward network (usually a two-layer MLP with a nonlinear activation) followed by another layer normalization and residual connection [33, 41, 56, 62]. The number of encoder layers determines the depth and capacity of the model. For example, BERT-Base uses 12 layers, BERT-Large uses 24 layers, and GPT-3 includes 96 layers [34, 40, 41].

The decoder layers make up the third main module of the transformer architecture. They are used in generative models and sequence-to-sequence frameworks to reconstruct original inputs, as in autoencoders, or to predict target outputs in supervised tasks based on the encoded representations [33, 35, 64, 71, 66, 62].

In encoder-decoder architectures such as the original transformer for machine translation, the decoder includes both masked self-attention and encoder-decoder cross-attention. Masked self-attention ensures that the model attends only to previously generated tokens, preserving the autoregressive nature of the generation process. The cross-attention mechanism enables the decoder to attend to the encoder's output representations [64, 71, 62]. This cross-attention allows the decoder to focus on relevant parts of the input sequence when producing each output token. In this process, the queries come from the decoder's current hidden state, while the keys and values are derived from the encoder's outputs [64, 71, 62].

Decoder-only architectures, such as the GPT series, remove the cross-attention layer and use only causal masked self-attention, where each token attends only to itself and earlier tokens in the sequence [33, 56, 44, 47]. In contrast, encoder-only models such as BERT eliminate the decoder and use the encoder's representations directly for downstream tasks like classification or token-level prediction, typically through simple task-specific output layers [41, 56, 72, 42, 73].

1.4.2 Pre-training and Fine-tuning Paradigm in Transformers

The training process for foundation models follows a two-stage approach known as "pre-train then fine-tune," which has become the standard method across many domains [33, 36, 56, 74, 72, 75, 76, 77]. In the pre-training stage, models learn general-purpose representations by training on large amounts of unlabeled data. They use self-supervised objectives that create learning signals directly from the data structure without the need for human labeling [33, 34, 56, 74, 42, 73, 78].

For language models, common pre-training objectives include masked language modeling (MLM) and causal language modeling (CLM). MLM, used by BERT and its variants, randomly masks 15% of input tokens and trains the model to predict the masked tokens using the surrounding context [34, 41, 42, 73, 79, 80, 81, 82]. CLM, also known as next-token prediction, is used by GPT models and trains the model to predict the next token in a sequence based on all previous tokens [34, 44, 45, 46, 83, 47, 84]. These self-supervised tasks allow models to learn complex linguistic features, such as syntax, semantics, and broader world knowledge, from the training data [41, 56, 42, 78].

In the case of vision transformers, pre-training can be either supervised, using large labeled image datasets such as ImageNet, or self-supervised, using approaches like masked autoencoding. In masked autoencoding, random patches of the input image (typically 75%) are hidden, and the model learns to reconstruct the missing pixel information from the visible patches and the learned latent representations [52, 53, 52, 55, 56, 49, 51]. The asymmetric encoder-decoder structure of MAE, which uses a lightweight decoder that operates only on mask tokens, makes it possible to train large vision models efficiently by processing only visible patches with the computationally heavy encoder [52, 54, 56].

After pre-training, models go through a fine-tuning stage, also known as transfer learning, to adapt the learned representations for specific downstream tasks using smaller labeled datasets [56, 85, 74, 72, 75, 76, 77, 86, 87, 88]. Fine-tuning starts by initializing the task-specific model with the pre-trained weights and then continuing training with supervised learning objectives designed for the target task [56, 72, 75, 76, 77].

There are several fine-tuning strategies that differ in how model parameters are updated. In full fine-tuning, all parameters are updated on the downstream task data, which gives the best performance but requires high computational resources [72, 76, 77, 88]. Layer-wise or selective fine-tuning freezes some layers, usually the early ones that capture general features, while updating others [76, 77, 88]. In contrast, feature extraction keeps the entire pre-trained model fixed and trains only a new task-specific head on top of the extracted representations [75, 77, 88].

More recent parameter-efficient fine-tuning methods, such as adapters, prompt tuning, and Low-Rank Adaptation (LoRA), add small trainable modules or make limited adjustments to the model while keeping most pre-trained parameters unchanged. This allows models to adapt efficiently with much lower computational cost [76, 88]. Overall, fine-tuning takes advantage of the general knowledge captured during pre-training and achieves better performance on downstream tasks compared to training from scratch, especially when labeled data is scarce [56, 89, 75, 77].

Beyond standard fine-tuning, foundation models show impressive few-shot and zero-shot learning abilities. These capabilities allow models to perform new tasks with little or no task-specific training data by using their pre-trained knowledge and in-context learning skills [34, 36, 90, 91, 92, 93, 94].

Zero-shot learning enables a model to complete tasks it was never directly trained for by using natural language instructions or task descriptions as prompts. The model relies only on its existing knowledge to generate suitable outputs [90, 93, 94]. For example, a language model can be prompted with "Translate the following English sentence to French:" followed by an English sentence, and it can produce the correct French translation even without explicit fine-tuning for translation, as long as it encountered similar examples during pre-training [94].

Few-shot learning builds on this idea by providing a small number of examples, usually between 1 and 10, within the prompt. This helps the model quickly adapt to a new task through in-context learning without changing its parameters [34, 90, 92, 93, 95]. GPT-3 demonstrated remarkable few-shot performance, achieving results comparable to or better than fine-tuned models on many benchmarks by conditioning on only a few examples in the prompt [34, 36]. These capabilities suggest that foundation models develop general problem-solving and meta-learning skills that go beyond task-specific training [34, 36, 91, 92].

1.5 Foundation Models in Biology and Genomics

In the field of biological sequence analysis and single-cell genomics, transformer-based foundation models have been adapted to learn representations from DNA sequences and gene expression data. These models draw direct parallels between natural language and cellular biology, where words correspond to genes, sentences to cells, and corpora to large atlases of tissues and conditions [96, 97, 98, 99, 100, 63, 101, 31, 102, 103].

Several important single-cell foundation models have been developed. scGPT [31] uses a generative pre-trained transformer trained on more than 33 million human cells with masked generative pre-training and causal masking, achieving state-of-the-art results in cell type annotation, batch correction, perturbation response prediction, and gene regulatory network inference [63, 31, 102, 104]. scBERT [105] applies a performer-based transformer architecture with gene2vec embeddings and masked gene expression prediction to learn gene and cell representations [63, 102, 106].

Geneformer [30] treats cells as sentences made up of rank-ordered gene tokens and learns context-aware gene embeddings using self-attention across millions of cells [101, 102]. CellPLM is trained on large-scale

single-cell data using masked language modeling objectives [102]. scFoundation [1] focuses on learning generalizable representations across multiple tissues and experimental conditions [101, 102]. CellFM, a foundation model with 800 million parameters trained on 100 million human cells, captures cellular diversity at an unprecedented scale [98, 101].

In addition to single-cell models, genomic foundation models such as the Nucleotide Transformer have been pre-trained on DNA sequences from multiple species, ranging from 50 million to 2.5 billion parameters. These models can learn regulatory patterns, predict variant effects, and transfer knowledge across species without task-specific supervision [107, 108, 109, 110, 111]. DNABERT applies bidirectional transformer pre-training to genomic sequences, enabling the capture of both local motifs and long-range dependencies within regulatory DNA [103, 109]. Enformer extends transformer architectures with larger receptive fields to predict gene expression and chromatin states from DNA sequences by attending to regulatory elements located up to 100kb away [112, 108].

These biological foundation models follow the same architectural principles used in language and vision transformers, employing token embeddings (for genes or nucleotides), positional encodings, stacked self-attention layers, and masked prediction objectives. However, they also include domain-specific adaptations such as gene expression binning strategies, conditioning embeddings that encode experimental metadata, and customized decoders for reconstruction tasks [99, 63, 31, 113, 102, 104, 114]. The use of foundation model approaches in single-cell biology offers great potential for accelerating cell type discovery, mapping developmental trajectories, predicting responses to perturbations, and advancing precision medicine [96, 115, 100, 63, 101, 31, 102, 114].

1.6 Limitations of Current Foundation Models in scRNA-seq

Although masked and autoregressive language modeling approaches have been highly successful in natural language processing and computer vision [34, 52, 41], applying these frameworks to scRNA-seq data presents major challenges. scRNA-seq datasets have several unique characteristics that make analysis difficult. They are extremely sparse because of technical dropout events, where gene expression is not detected even though mRNA is actually present [116, 117, 118, 119, 120, 121]. The data are also highly dimensional, often spanning tens of thousands of genes with complex patterns of co-expression [117, 122, 123, 124], and they contain considerable technical noise caused by random sampling, batch effects, and platform-specific biases [125, 117, 126, 127]. Together, these factors make downstream analyses such as cell-type classification particularly difficult [128, 129, 130, 131, 132].

The proportion of zero values in scRNA-seq data can reach up to 90% [116, 118, 133]. These zeros may result from either true biological absence of gene expression or from technical artifacts such as dropout events during library preparation, reverse transcription, or sequencing [116, 118, 118, 119, 120, 134]. Distinguishing between biological zeros and technical dropouts remains a long-standing issue, made worse by inconsistent use of the term “dropout” in the literature, which has been applied to refer to non-biological zeros, all zeros, or even moderate gene expression in a subset of cells [116, 118, 120].

Additionally, scRNA-seq data suffer from severe curse-of-dimensionality effects. As noise accumulates across thousands of gene dimensions, true biological similarities between cells become harder to detect. This reduces clustering accuracy and weakens the statistical power of differential expression analyses [117, 126, 118, 122, 123].

When masked language modeling is applied to scRNA-seq data, uniform random masking, the standard approach used in BERT and related models [73, 42, 41], often selects genes that carry little biological information and do not contribute meaningfully to distinguishing cell types [135, 136, 137, 138, 139]. This issue arises because information content is distributed very unevenly across genes. A small subset of highly variable genes (HVGs) captures most of the biologically relevant variation that differentiates cell types and states [137, 138, 140, 141, 142], while most other genes show little variance, are expressed at low levels in all cells, or are dominated by technical noise rather than true biological signal [136, 137, 138, 143].

As a result, when a transformer-based foundation model applies a uniform 15% masking rate across all genes, a strategy borrowed from natural language processing, where words tend to carry more evenly distributed information [73, 42, 144], most of the masked positions correspond to zero values or non-variable genes rather than informative HVGs that define cell identity [135, 114, 136]. This mismatch greatly reduces the effectiveness of training, since most reconstruction targets provide little useful signal for learning meaningful gene-gene relationships and cell states [135, 114, 143, 145].

Empirical studies have shown that scRNA-seq foundation models trained with standard masked language modeling objectives often perform no better than models that simply predict the average expression value for each gene. This indicates that such pre-training contributes little beyond memorizing overall expression levels [114, 63]. Furthermore, uniform masking prevents the model from focusing on complex co-expression patterns among marker genes and regulatory networks that define cellular identity [135, 139, 137, 146]. Consequently, applying random masking directly to scRNA-seq data can fail to improve performance and may even reduce it compared to simpler baseline methods, as the model spends its capacity reconstructing uninformative zeros and housekeeping genes instead of the critical HVGs [114, 63, 143, 144].

Beyond reduced learning efficiency, the randomness in traditional masking introduces a major limitation for scientific interpretability and mechanistic understanding of foundation models in biological research [147, 148, 149, 150, 151, 152]. Because masking is applied randomly and changes across training epochs through dynamic masking strategies [73, 147, 148], it becomes extremely difficult to study which genes are masked or to systematically analyze how masking affects different components of a foundation model [147, 149, 153, 154].

This lack of control makes it impossible to determine which genes or gene programs the model learns to reconstruct accurately, to trace attention patterns linked to biologically meaningful gene sets, or to perform targeted ablation studies that test how masking specific functional modules (such as cell cycle, immune response, or metabolic pathway genes) influences downstream performance [149, 153, 155, 139, 151, 154]. Such opacity is a serious problem in biological applications, where researchers and clinicians need mechanistic insight into model behavior. They must be able to interpret which genes drive cell type predictions,

how developmental trajectories are represented, and whether learned relationships correspond to known regulatory mechanisms [150, 153, 155, 139, 151, 152].

The inability to control or analyze masking strategies also prevents the use of biologically motivated masking schemes that could enhance both model performance and interpretability [147, 148, 135, 156]. For instance, focusing masking on highly variable genes, antibody complementarity-determining regions, or non-templated sequences has been shown to improve computational efficiency and biological relevance [147, 148, 135, 157, 156]. Other domains have explored more advanced masking methods, including adaptive masking rates, gene-importance-weighted sampling, multi-scale sub-vector completion, and spatially structured masking. These strategies show promise for improving scRNA-seq foundation models but remain largely unexplored because most current methods still rely on uniform random masking inherited from NLP [147, 148, 158, 157, 156, 143].

1.7 Research Gap and Motivation

Single-cell foundation models have advanced rapidly through large-scale pretraining and transformer-based architectures, yet the pretraining pipelines used in nearly all existing models continue to rely on uniform random masking. This approach assumes that all genes contribute equally to the reconstruction task, despite the extreme sparsity, zero inflation, and heterogeneous expression patterns characteristic of scRNA-seq datasets. Such a mismatch between the masking strategy and the structure of the data often leads models to reconstruct uninformative or non-expressed genes, limiting the quality of learned representations and contributing to instability when models are transferred across biological contexts [150, 152, 154, 99].

In evaluating several widely adopted single-cell foundation models, we observe these limitations manifest across diverse datasets and biological settings. Models tend to perform well on abundant or transcriptionally distinct cell types but exhibit marked degradation on rare populations or clinically challenging states. While these observations arise from empirical benchmarking, they ultimately reflect a broader issue: the masking objectives used during pretraining do not guide models toward biologically meaningful variation.

Recent advances such as Self-guided Masked Autoencoders (SMA) [159] demonstrate the promise of attention-driven masking in other domains. However, these methods rely on computationally intensive teacher–student architectures and are designed for dense, low-dimensional inputs, making them poorly suited for high-dimensional, sparse, 20k-token scRNA-seq data. Moreover, existing approaches provide limited insight into how masking information flows through different parts of transformer models, an important consideration for encoder–decoder architectures commonly used in single-cell genomics.

These limitations indicate the need for a masking strategy that is biologically informed, computationally efficient for full-length gene-expression inputs, and compatible with existing single-cell foundation model designs without requiring additional large auxiliary networks.

1.8 Proposed Approach

This work is organized around two complementary objectives aimed at advancing the development and assessment of single-cell foundation models. The first objective is to establish a systematic and controlled evaluation framework for existing single-cell FMs. To this end, we benchmark four widely used models including Geneformer[160], scBERT[105], scFoundation[1], and scGPT[31], across three biologically diverse datasets and under standardized fine-tuning regimes. This evaluation examines generalization across tissues, clinical states, and rare cell populations, providing a comprehensive characterization of how current FMs behave under realistic single-cell settings.

The second objective is to explore how improvements to pretraining strategies may enhance foundation model performance. For this purpose, we introduce Multinomial Attention Masking (MAM), a masking framework specifically designed for the sparsity and heterogeneity of scRNA-seq data. In contrast to uniform random masking, MAM employs a set of trainable latent vectors that attend over gene embeddings to infer gene importance. These attention scores are converted into a multinomial distribution from which masked positions are sampled, creating a dynamic, biologically informed masking pattern. The resulting latent summary is integrated into the backbone through four context-injections (encoder-only, decoder-only, shared adapter, or separate adapters), allowing masking information to be incorporated flexibly across transformer architectures.

Together, these two components provide a broad perspective on the current capabilities of single-cell foundation models and potential avenues for improving their pretraining dynamics.

Contribution: This work makes the following contributions:

- **Benchmarking Framework:** We present a unified and controlled evaluation of four major single-cell foundation models across three datasets, providing a detailed analysis of their performance across tissues, disease states, and rare cell types.
- **Analysis of Generalization Behavior:** Our study shows that model performance is uneven: the models work well on common cell types but struggle on rare populations and on difficult clinical states.
- **Multinomial Attention Masking:** We introduce MAM, a dynamic and biologically informed masking strategy that uses latent cross-attention and multinomial sampling to prioritize informative gene positions during pretraining.
- **Performance Improvements Across Datasets:** Through extensive experiments, we show that MAM consistently enhances cell-type annotation performance and reduces representational degradation compared with uniform random masking, improving both partial finetuning and linear probing outcomes.
- **Biological Validation of Masking Behavior:** We demonstrate that MAM preferentially masks highly expressed and functionally meaningful genes, with enriched pathways aligning with known immune and lineage-specific processes, indicating that MAM captures biologically relevant structure rather than relying on arbitrary masking patterns.

2 Literature Review

2.1 Classical scRNA-seq Analysis Methods

Single-cell RNA sequencing data analysis follows a well-established computational pipeline that has been refined through years of research. The standard workflow begins with quality control procedures to remove low-quality cells and filter out genes with very low expression [161, 9, 11]. Quality control metrics typically include the total number of detected genes per cell, library size, and the percentage of reads mapping to mitochondrial genes, which indicates cell damage or death [162, 11]. Tools such as Scater provide comprehensive frameworks for performing rigorous quality control across different scRNA-seq protocols [161].

After quality control, normalization is essential to correct for technical variation caused by differences in sequencing depth, capture efficiency, and amplification bias [163, 164]. Simple methods such as library size normalization scale counts by the total counts per cell, while more advanced techniques like scran's deconvolution approach compute cell-specific size factors that are robust to differences in cell type composition [165, 163].

The sctransform method, based on regularized negative binomial regression, effectively models technical noise and has shown strong performance in preserving biological variation while removing artifacts [166]. Comprehensive comparisons of normalization methods demonstrate that the chosen method significantly affects downstream results, with top-performing approaches showing higher consistency with external validation data [163].

Feature selection is a crucial step that determines which genes are used for dimensionality reduction and clustering. The aim is to identify highly variable genes (HVGs) that capture biological differences while excluding genes dominated by technical noise [167, 168]. Classical approaches model the relationship between mean expression and variance, selecting genes with significantly higher variance than expected from noise models [167, 9].

Modern implementations in Seurat use variance-stabilizing transformations or local polynomial regressions to identify genes with standardized variance above a threshold [14, 13]. Other methods, such as M3Drop, model dropout rates, while ROGUE selects genes based on expression entropy [169, 170]. Benchmarking studies show that performance varies depending on the dataset and task, but highly expressed genes often provide enough resolution to distinguish cell types [171, 172].

Dimensionality reduction addresses the curse of dimensionality by projecting high-dimensional expression data into a lower-dimensional space while preserving key structures [173, 174]. Principal component analysis (PCA) remains the standard linear method, capturing directions of maximum variance [11].

For visualization, nonlinear methods such as t-distributed stochastic neighbor embedding (t-SNE) and uniform manifold approximation and projection (UMAP) are widely used [175, 176]. t-SNE is effective at revealing local structure and fine-grained clusters, while UMAP better preserves global relationships and

scales efficiently to large datasets [177, 174]. Comparative studies find that UMAP provides higher stability across parameter settings, while t-SNE achieves slightly better separation of known cell populations [174]. More recent deep learning methods, such as scvis, use autoencoders to learn nonlinear low-dimensional representations that can capture both linear and nonlinear relationships [178].

Cell clustering groups transcriptionally similar cells to identify distinct types and states. Graph-based clustering has become the dominant approach, where k-nearest neighbor (KNN) graphs are built from cell similarity measures and then partitioned into communities [11, 179]. The Louvain algorithm optimizes modularity to maximize within-community density, while the Leiden algorithm improves upon it by ensuring better-connected clusters and avoiding disconnected communities [180, 181]. Both methods include a resolution parameter that controls cluster granularity, with higher values producing more detailed subclusters [181].

Alternative techniques include hierarchical clustering, as in SC3, which combines multiple clustering solutions for consensus results [182], and density-based methods such as DBSCAN [179]. Comparative evaluations show that graph-based methods, especially Leiden and its variants, yield reliable and interpretable results across diverse datasets [183].

Cell type annotation assigns biological labels to clusters. Traditionally, this has been done by manually examining marker gene expression [184, 185], but manual annotation is time-consuming and subjective. Automated methods have therefore gained popularity. Reference-based tools such as SingleR transfer labels from annotated reference datasets by correlating query profiles with known cell types [186].

Marker-based methods like CellAssign use probabilistic models that incorporate curated marker gene sets [187]. Supervised machine learning approaches train classifiers on labeled reference data, including models based on support vector machines (scPred), random forests (singleCellNet), and neural networks (ACTINN) [188, 189, 190]. The growing availability of large-scale cell atlases has further enabled transfer learning approaches that leverage representations learned from millions of cells [191].

2.2 Deep Learning Methods for scRNA-seq Analysis

The use of deep learning in scRNA-seq analysis has transformed the field by enabling more advanced modeling of the complex structure present in single-cell data [192, 193]. These approaches take advantage of neural networks' ability to learn hierarchical representations and capture nonlinear relationships that traditional methods often cannot model effectively. Autoencoders have become one of the most fundamental architectures for unsupervised representation learning in scRNA-seq analysis [194, 195].

These networks learn compact latent representations by reconstructing input gene expression profiles through an encoder-decoder structure. Early work such as AutoImpute showed that autoencoders can accurately impute dropout events by learning the underlying distribution of gene expression without relying on explicit statistical models [194].

Studies investigating optimal autoencoder designs for scRNA-seq found that deeper and narrower architectures using sigmoid or tanh activations perform better than shallower networks using ReLU activations,

which contrasts with common practices in computer vision [196]. Regularization techniques such as weight decay and dropout were also shown to be crucial for improving imputation accuracy and enhancing downstream analyses including clustering and differential expression [196].

Variational autoencoders (VAEs) extend the standard autoencoder by introducing probabilistic latent variables and explicit generative modeling [197, 29]. The scVI framework applies VAEs to model scRNA-seq counts using a zero-inflated negative binomial likelihood, allowing the model to capture both biological variation and technical dropout [29]. The probabilistic nature of scVI supports uncertainty quantification and enables multiple downstream applications such as dimensionality reduction, batch correction, and differential expression testing [29, 198].

Further developments include biVI, which integrates biophysical modeling; LDVAE, which provides interpretable linear decoders; and totalVI, which extends the model to multi-modal data [199, 200, 201]. However, recent studies have questioned whether explicit zero-inflation modeling significantly improves performance compared to simpler negative binomial models for modern high-quality datasets [202].

Graph neural networks (GNNs) have also shown strong performance in scRNA-seq analysis by directly modeling cell-cell relationships and gene regulatory networks [203, 204]. The scGNN framework represents cell relationships as graphs and aggregates information from neighboring cells using graph convolutions, enabling joint imputation, clustering, and network inference [203].

It uses a left-truncated mixture Gaussian model to describe heterogeneous gene expression and iteratively refines both the graph structure and learned embeddings [203]. Other GNN-based models include GNNLink, which infers gene regulatory networks using attention-weighted message passing, and scGraph, which performs cell type annotation by integrating gene-gene interaction networks [205, 206].

Contrastive learning and self-supervised methods have recently gained importance as effective strategies for learning robust cell representations without large labeled datasets [207, 208]. The CLEAR framework uses contrastive learning by generating augmented views of each cell through random noise and training the model to produce similar embeddings for these views while separating different cells [207]. This design enforces the principle that functionally similar cells should lie close together in the embedding space. Later methods combine instance-level and cluster-level contrastive objectives; for example, ScCCL employs momentum encoders and dual-level contrastive loss to enhance clustering accuracy [208].

More advanced frameworks such as scAMAC incorporate multi-scale attention mechanisms that combine features from encoder, hidden, and decoder layers to capture cellular characteristics at different resolutions [209]. Comparative studies indicate that contrastive learning methods can outperform traditional VAE-based approaches, especially for cell type annotation tasks across diverse datasets [210].

2.3 scRNA-seq Foundation Models and Transformers

The success of transformer architectures in natural language processing and computer vision has inspired their adaptation to single-cell transcriptomics [30, 31]. These foundation models use self-supervised pre-

training on large single-cell datasets to learn general representations of cellular states that can be transferred to many downstream tasks. The transformer architecture, first introduced in "Attention Is All You Need," consists of three main components: token embeddings with positional encodings, stacked encoder layers with multi-head self-attention, and optional decoder layers for generative modeling [211].

In the context of scRNA-seq, genes act as tokens similar to words in language models, cells correspond to sentences, and large single-cell atlases serve as training corpora [30, 31]. The self-attention mechanism allows each gene to attend to all other genes within a cell, capturing complex co-expression patterns and gene-gene dependencies without assuming independence [211, 212].

scGPT is one of the first large-scale foundation models developed for single-cell analysis [31]. It was pre-trained on more than 33 million human cells using masked generative pre-training that combines random and causal masking strategies. The model applies gene expression binning to convert continuous expression values into discrete categories and integrates gene, cell, and condition embeddings [31]. scGPT incorporates Flash Attention mechanisms to improve computational efficiency and learns a CLS token to summarize the cellular state [31]. When evaluated across multiple downstream tasks such as cell-type annotation, batch correction, perturbation prediction, and gene regulatory network inference, scGPT achieved state-of-the-art results, especially for modeling responses to perturbations [31].

scBERT adapts the bidirectional encoder representations from transformers (BERT) framework to scRNA-seq data by including gene2vec embeddings that encode gene identity and expression embeddings representing binned expression levels [105]. Its architecture uses Performer layers to efficiently process long gene sequences and is pre-trained using masked language modeling on the Panglao reference database [105]. However, later analyses questioned the effectiveness of the learned representations. Surprisingly, pre-training performance remained high even after removing gene identity information, suggesting that the model mainly captures simple expression statistics rather than complex gene-gene interactions [213].

Geneformer takes a different approach by representing cells as sequences of rank-ordered gene tokens instead of using fixed gene orderings [30]. This approach allows the model to learn context-aware gene embeddings that reflect positional importance within each cell's expression profile [30]. Pre-trained on tens of millions of human cells from diverse tissues, Geneformer performed strongly in gene network inference and in silico perturbation prediction [30]. The rank-ordering strategy also improves computational efficiency by focusing attention on highly expressed genes and naturally down-weighting lowly expressed or dropout genes. Several other foundation models have explored alternative architectures and training strategies.

CellPLM adapts protein language model architectures with masked language modeling objectives [214]. scFoundation trains on large single-cell datasets to learn generalizable representations across tissues and conditions [1]. CellFM scales to 800 million parameters trained on 100 million human cells, making it one of the largest single-cell foundation models so far [215]. The Universal Cell Embedding (UCE) model extends single-cell foundation modeling to cross-species contexts, achieving zero-shot generalization to new organisms and enabling the identification of rare cell types through its learned embeddings [216].

Despite these advances, recent evaluations have revealed important limitations in scRNA-seq foundation models [213, 102]. Systematic benchmarking has shown that such models often do not consistently outperform simpler baseline approaches, particularly for cell-type annotation [213, 102]. Notably, removing pre-training from scBERT caused no major decline in fine-tuning performance, suggesting that pre-training may not capture the expected rich biological representations [213].

Studies on batch correction found that foundation models perform worse than specialized methods such as Harmony when dealing with strong batch effects [217]. Furthermore, even simple logistic regression models trained on highly variable genes remain competitive and sometimes outperform foundation models in few-shot learning settings [213]. These findings highlight the need for more rigorous benchmarks and a deeper understanding of what biological representations foundation models truly learn.

2.4 Pre-training and Fine-tuning Paradigms

Foundation models generally follow a two-stage "pre-train then fine-tune" training framework that has become standard across multiple domains [212, 218]. This approach allows models to first learn general-purpose representations from large unlabeled datasets and then adapt these representations to specific downstream tasks with limited labeled data. During the pre-training stage, models use self-supervised objectives that create learning signals directly from the structure of the data, removing the need for human annotation [212, 218].

In language modeling, two main pre-training strategies are used: masked language modeling (MLM) and causal language modeling (CLM). MLM, introduced with BERT, randomly masks about 15% of the input tokens and trains the model to predict the masked tokens using information from surrounding unmasked tokens [212]. Typically, 80% of the selected tokens are replaced with a [MASK] token, 10% with random tokens, and 10% are left unchanged [212]. This setup helps the model learn strong contextual representations while preventing it from simply associating [MASK] with prediction targets [212].

CLM, also known as next-token prediction or autoregressive modeling, trains the model to predict the next token in a sequence based on all previous tokens [219, 218]. GPT and related models use this objective to generate coherent text by learning the conditional probability of each token given its context [219, 218].

For vision transformers, pre-training can be done either through supervised learning on large labeled datasets such as ImageNet or using self-supervised methods like masked autoencoding [220, 52]. Masked Autoencoders (MAE) randomly hide a large proportion (typically 75%) of image patches and train the model to reconstruct the missing pixel values [52]. The asymmetric encoder-decoder design allows only visible patches to pass through the full encoder, reducing computational cost, while a lightweight decoder reconstructs the full image from the encoded visible patches and learnable mask tokens [52]. This design efficiently trains large vision models while encouraging the encoder to learn meaningful, generalizable representations [52].

Fine-tuning adapts the pre-trained model to a particular downstream task using smaller labeled datasets [212, 221]. The model's weights are initialized from the pre-trained parameters and then updated using

supervised objectives specific to the task [212, 221]. Several fine-tuning strategies are commonly used. Full fine-tuning updates all parameters but is computationally demanding [212]. Layer-wise fine-tuning freezes early layers and updates only later ones [221], while feature extraction keeps the entire pre-trained model fixed and trains only new output layers [221].

More recently, parameter-efficient fine-tuning methods such as adapters, prompt tuning, and Low-Rank Adaptation (LoRA) have been developed. These methods introduce small trainable modules or parameter updates while keeping most pre-trained weights frozen, reducing both computational and memory requirements [222, 223, 224].

The effectiveness of pre-training is rooted in transfer learning, where patterns learned from large and diverse datasets can be transferred to related tasks, particularly when labeled data are limited [225, 226].

Empirical evidence consistently shows that fine-tuning pre-trained models leads to better performance than training from scratch, especially in data-scarce settings [212, 218]. However, recent studies highlight that the benefits of pre-training are not universal. With sufficiently strong data augmentation, pre-training may offer diminishing returns or even reduce performance [227]. These results suggest that the success of pre-training depends strongly on the alignment between pre-training data, augmentation methods, and the nature of the downstream task.

2.5 Few-Shot and Zero-Shot Learning

Beyond standard fine-tuning, foundation models show impressive few-shot and zero-shot learning abilities that allow them to perform new tasks with little or no task-specific training [218, 228]. These capabilities indicate that large-scale pre-training equips models with general problem-solving skills that go beyond memorizing training data. Zero-shot learning enables models to handle tasks they were not explicitly trained for by using natural language instructions or task descriptions as prompts [228]. The model relies entirely on its pre-trained knowledge and the structure of the prompt to generate the correct output [228].

For example, GPT-3 can translate text by simply being prompted with "Translate the following English sentence to French:" followed by the input sentence, achieving reasonable translation results without any fine-tuning on translation datasets [218]. Similarly, vision-language models such as CLIP perform zero-shot image classification by learning shared embeddings for images and text, allowing them to classify images based on textual descriptions of classes [228]. The key principle is that when trained on large and diverse datasets, models can generalize to new task formats by recognizing familiar patterns and structures from their pre-training data.

Few-shot learning builds on the zero-shot concept by providing a small number of demonstration examples (usually between 1 and 10) within the prompt to guide model behavior [218]. This process, known as in-context learning, does not require updating the model's parameters but instead conditions the model's predictions on the examples provided [218].

GPT-3 showed that performance improves significantly with more in-context examples, often matching or surpassing fine-tuned models on a range of benchmarks despite using only a few demonstrations [218]. This few-shot capability suggests that large models develop a form of meta-learning during pre-training, learning not only task-specific representations but also the ability to learn from new examples and adapt to unfamiliar tasks [218].

The mechanisms behind few-shot and zero-shot learning remain an active area of investigation. Research suggests that these abilities arise from a combination of pattern recognition, compositional reasoning, and implicit task understanding driven by the structure of the prompt [229]. The quality of demonstration examples has a strong influence on performance, with semantically relevant examples producing better results than random ones [230]. Likewise, the clarity and structure of the prompt itself play a crucial role, as well-designed instructions often lead to much stronger model performance [231].

2.6 Attention-Based Masking and Adaptive Strategies

The design of masking strategies for self-supervised pre-training has become a key factor influencing the quality of learned representations [212, 52]. Although random uniform masking remains the most widely used approach, recent studies have introduced more advanced adaptive and attention-guided methods. In random masking strategies, tokens are selected for masking with equal probability, assuming that all tokens contribute equally to learning [212].

In BERT, for example, 15% of tokens are randomly masked to generate diverse training signals across different masked configurations [212]. Vision transformers extend this idea by masking a large proportion (75%) of image patches, taking advantage of the high redundancy present in natural images [52]. However, uniform masking may not be ideal when information is unevenly distributed across tokens, such as in biological sequences where only a small subset of informative genes carry most of the discriminative information.

Attention-based masking strategies make use of model attention patterns to identify which tokens are most important for masking [232]. The main idea is that tokens receiving high attention from others likely contain more meaningful information, making them better reconstruction targets [232].

The Self-Guided Masked Autoencoder (SMA) applies this principle by using attention maps from a teacher model to guide the masking process in a student model, allowing it to identify informative regions in a domain-independent way [159]. While effective, this approach introduces significant computational overhead due to the teacher-student framework and has mostly been tested on dense, low-dimensional datasets.

Curriculum masking gradually increases the difficulty of the masking process during training, starting with easier tokens and progressively including harder ones [233]. The Curriculum Masking-based Gene Masking Strategy (CM-GEMS) applies this idea to genomic transformer models by using Pointwise Mutual Information (PMI) to measure token difficulty and implementing a time-dependent masking schedule [233]. This approach has shown that models trained with curriculum masking can achieve 90% of state-of-the-art performance while using 10× fewer training steps compared to standard random masking [233].

Other masking methods focus on specific sequence regions (such as antibody complementarity-determining regions), apply multi-scale sub-vector masking, or use gene-importance-weighted sampling to target biologically relevant features [234, 235].

Masked attention mechanisms extend traditional attention by explicitly controlling which tokens can interact during attention computation [211]. Causal masking ensures autoregressive left-to-right processing by preventing tokens from attending to future positions [211]. Spatial masking limits attention to specific regions or objects, which is commonly used in vision tasks like object detection and segmentation [236].

Structured masking introduces fixed interaction patterns such as sliding windows or hierarchical attention to improve efficiency and embed assumptions about the problem structure [237]. Together, these mechanisms show that carefully designed attention constraints can enhance both computational efficiency and model performance by incorporating domain-specific knowledge into the architecture.

2.7 Challenges in scRNA-seq Data

Single-cell RNA sequencing (scRNA-seq) data introduces unique computational challenges that set it apart from other domains where transformers have been successful [11, 238]. These challenges have important implications for the design and evaluation of foundation models in single-cell analysis. Dropout events and zero-inflation are defining characteristics of scRNA-seq data, where zero values often account for 90% or more of the entire expression matrix [239, 120]. These zeros can arise from both true biological absence of expression and technical failures in RNA capture, reverse transcription, or sequencing [239].

Distinguishing between biological zeros and technical dropouts remains a difficult and debated issue, and terminology across studies is often inconsistent [239]. Earlier research suggested that explicitly modeling zero-inflation using zero-inflated negative binomial distributions could improve analysis [240], but newer evidence indicates that modern, high-quality protocols may not show significant zero-inflation beyond what standard negative binomial models already capture [202]. Regardless of the cause, the extreme sparsity of scRNA-seq data complicates the detection of biological patterns and intensifies the curse of dimensionality, making it difficult to separate meaningful biological variation from noise [120].

The uneven distribution of information across genes adds another major challenge for uniform random masking approaches [167, 9]. Highly variable genes (HVGs) capture most of the biologically meaningful variation that defines cell types and states, whereas most genes show little variation or are mainly influenced by technical noise [167, 9].

Recent benchmarking studies have found that selecting highly expressed or highly variable genes improves clustering resolution compared to methods that treat all genes equally [171]. Because of this uneven information distribution, the typical uniform 15% masking rate used in language model pre-training—where words carry more evenly distributed information—often masks uninformative genes in scRNA-seq data [213]. As a

result, most reconstruction targets offer weak learning signals, and models may achieve high pre-training accuracy by simply predicting average expression levels rather than learning complex gene-gene relationships [213].

Batch effects are another major source of technical variation, caused by differences in experimental protocols, library preparation, sequencing platforms, cell isolation procedures, and processing times [241, 242]. These variations can be as large as or even larger than true biological differences, leading cells of the same type from different batches to appear more distinct than cells of different types from the same batch [241].

Traditional batch correction methods include linear approaches such as Combat, mutual nearest neighbor (MNN) algorithms, and integration techniques based on canonical correlation analysis (Seurat) or adversarial learning (scVI, LIGER) [243, 244, 14]. A large-scale comparison of batch correction tools found that Harmony consistently performs best across multiple metrics, effectively removing batch effects while preserving biological structure [217].

However, some methods such as MNN, SCVI, and LIGER have been reported to introduce artifacts or distort data structure during correction [217]. Foundation models face similar challenges, as evaluations show that pre-trained models often perform poorly in the presence of strong batch effects compared to specialized integration methods [241].

The curse of dimensionality is particularly severe in scRNA-seq data due to the combination of extremely high dimensionality (typically 20,000–30,000 genes), high noise, and relatively small sample sizes compared to the number of model parameters [173]. As dimensionality increases, standard distance metrics lose their discriminative power, and the likelihood of spurious correlations grows [173].

Feature selection can reduce dimensionality but cannot fully solve the problem, as even after selecting 2,000–5,000 HVGs, the data remain highly complex [11]. Deep learning methods must therefore balance sufficient model capacity to capture biological complexity with the need to avoid overfitting to noise [192]. Dimensionality reduction techniques such as PCA or learned embeddings help alleviate these challenges but introduce their own trade-offs between preserving global and local structure [173, 174].

3 Method

In the first objective of our study, we evaluate four off-the-shelf single-cell RNA-seq foundation models (Geneformer [30], scBERT [105], scFoundation [1], and scGPT [31]) on three different datasets. The goal of this step is to see how well these pretrained models work on data they have not seen before, and to understand whether any performance issues come from biases in their original pretraining.

For each model, we add a linear classification head and finetune it using the default settings and hyperparameters provided by the authors, so that our comparison is fair and consistent. After completing the evaluation on all three datasets, we then focus more deeply on the Acute Myeloid Leukemia (AML) dataset. We choose AML because it represents a rare and clinically important disease where many foundation models are expected to struggle. By focusing on AML, we can examine how each model behaves in a real low-prevalence and high-complexity setting.

In the next objective of our research, we introduce a novel masking approach, which uses the Multinomial Attention Masking (MAM) algorithm, and it can be easily added to off-the-shelf FMs. We use learnable parameters for learning the important features. We additionally propose novel approaches such as adding a multinomial attention masking unit and a cross-attention context injection strategy, to make our approach appropriate for sparse data in scRNA-seq.

3.1 Dataset

The single-cell RNA datasets contain a large, sparse matrix including information about gene activities across different cells. The gene activity values can be from 0 to large floating-point numbers, indicating how the gene is expressed. The higher the value is, the more expression is witnessed of that gene. In this study, we used three publicly available scRNA-seq datasets. Hematopoietic Niche scRNA-seq [3] and PBMC68K [1], and Adult Heart dataset [245]. For every dataset, we split the data into training, validation, and test partitions and apply standard preprocessing steps (e.g., normalization, gene filtering, or ranking transformations).

3.1.1 Description of Individual Datasets

Hematopoietic Niche scRNA-seq dataset (or Acute Myeloid Leukemia(AML)): The dataset is introduced by [3]. The dataset contains approximately 350,000 single cells collected from around 50 human donors, with each cell measured across 16,000 genes. Because the data come from multiple sources and include different donors, tissues, and disease conditions, the final dataset covers a wide range of biological variation that is useful for evaluating model robustness. The dataset provides two key types of labels: cell types and clinical timepoints. The clinical timepoints represent four biological states (Healthy, AML Diagnosis, AML Post-treatment, and AML Relapse) which together capture how the hematopoietic system changes as AML develops and progresses.

[1] https://support.10xgenomics.com/single-cell-gene-expression/datasets/

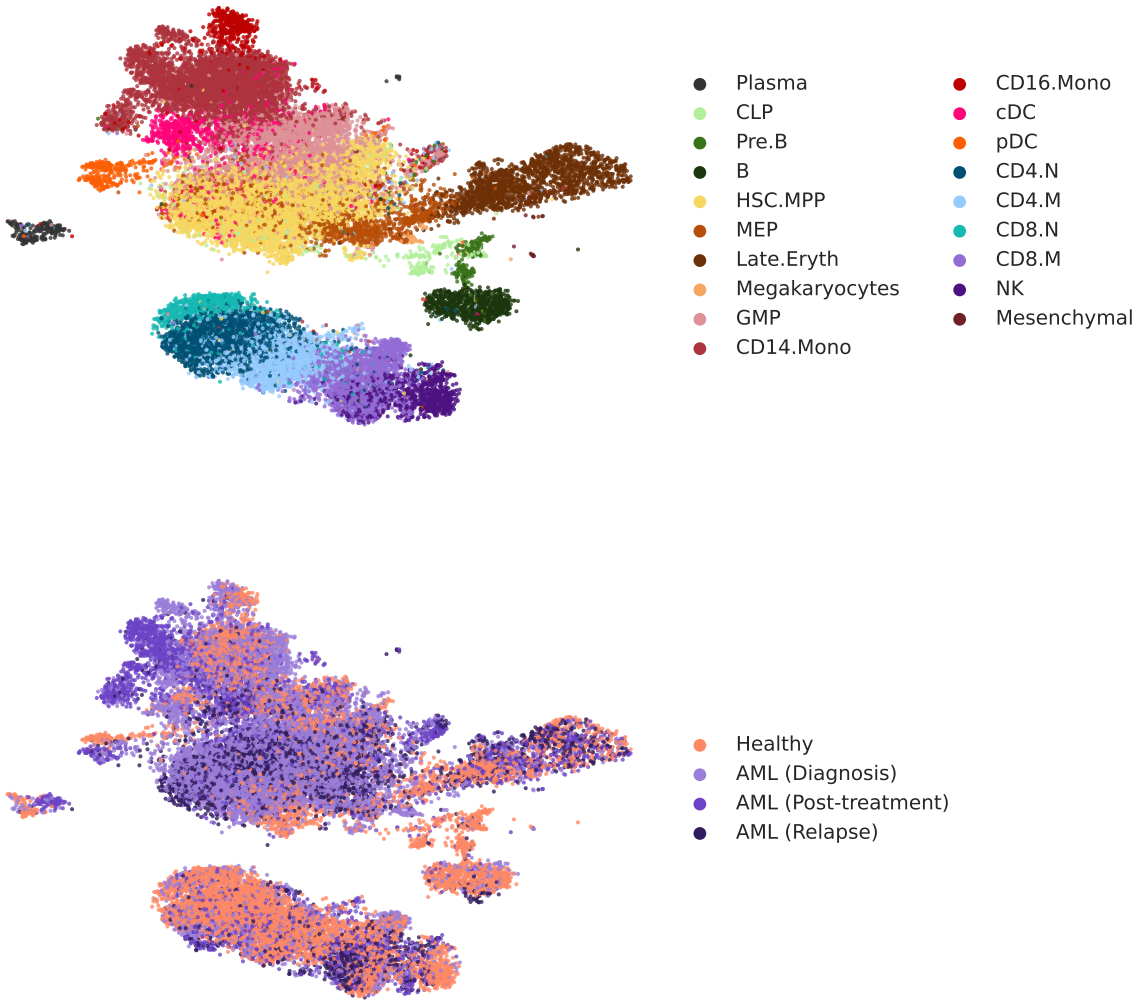

Figure 2: UMAP visualization of Hematopoietic Niche scRNA-seq (Acute Myeloid Leukemia) [3]. Each point represents a single cell colored by (top) cell type and (bottom) clinical timepoint.

The dataset also contains a diverse set of hematopoietic and immune cell types, including common populations such as B cells and Monocytes, as well as rare and clinically important populations like CLP, Pre.B, and HSC/MPP. These cell types vary widely in abundance, which makes the dataset especially useful for studying how models behave on both well-represented and underrepresented populations [3].

Figure 2 shows a UMAP visualization of the dataset, where each point represents a single cell. The top panel is colored by cell type, and the bottom panel is colored by clinical timepoint. This figure highlights the strong heterogeneity of the dataset across both biological and disease-related axes. It also visually illustrates the separation (and in some cases overlap) between different AML states and healthy samples. These characteristics make the dataset an effective benchmark for examining performance gaps, bias, and failure modes across both cell types and disease stages.

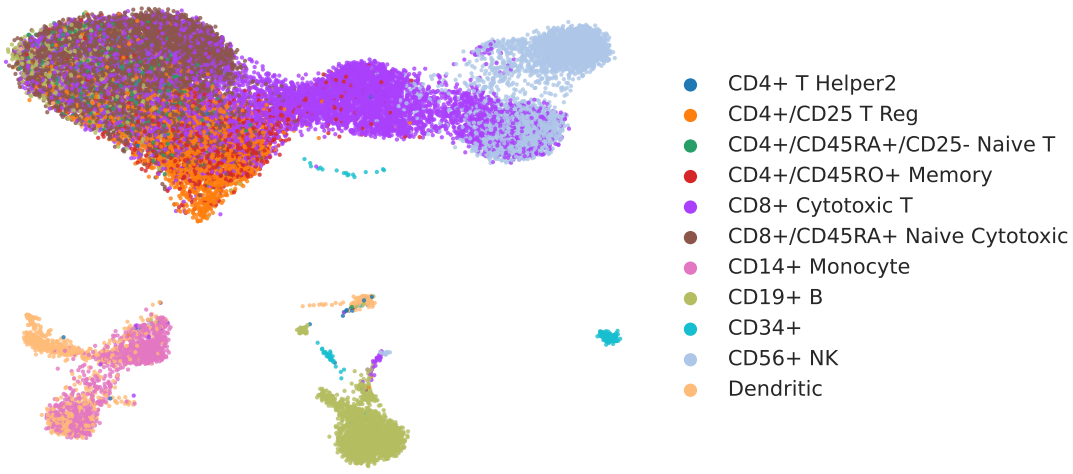

Figure 3: UMAP visualization of the PBMC68K dataset, where each point represents a single cell colored by its cell type.

PBMC68K dataset: The PBMC68K dataset [246], released by 10x Genomics [2], contains about 68,000 peripheral blood mononuclear cells (PBMCs) collected from healthy human donors. Each cell is measured across roughly 17,000 genes using a droplet-based scRNA-seq platform. The dataset includes major immune cell types such as B cells, NK cells, Monocytes, and Dendritic cells, along with basic gene-level metadata. Because the dataset is clean, well-annotated, and comes from healthy individuals, it is often used as a standard benchmark for testing clustering, representation learning, and reconstruction methods in single-cell analysis. It provides a good reference point for understanding how models behave on a typical immune system before moving on to more complex or disease-specific datasets.

Adult Heart dataset: This dataset [245] integrates single-cell and single-nucleus RNA sequencing data from both fetal and adult human hearts. The fetal component consists of samples from seven hearts collected between 6 and 10 weeks post-conception, with most tissues dissected into base and apex regions prior to sequencing. All fetal samples were processed using the 10x Genomics scRNA-seq platform. To provide a comprehensive representation of cardiac cell populations, these data were combined with adult heart samples from multiple anatomical regions of six healthy individuals. The adult profiles were generated using a combination of 10x scRNA-seq and 10x snRNA-seq technologies.

Figure 4 shows a UMAP visualization of the dataset, where each point represents a single cell or nucleus colored by its annotated cell type. Distinct clusters such as cardiac muscle cells, endothelial cells, fibroblasts, smooth muscle cells, pericytes, and fetal cardiomyocytes can be clearly seen. The mixture of fetal and adult populations, along with the variety of cardiac lineages, highlights the biological richness of this dataset and makes it a useful benchmark for studying model performance on both developmental and adult human tissues.

[2] https://support.10xgenomics.com/single-cell-gene-expression/datasets/

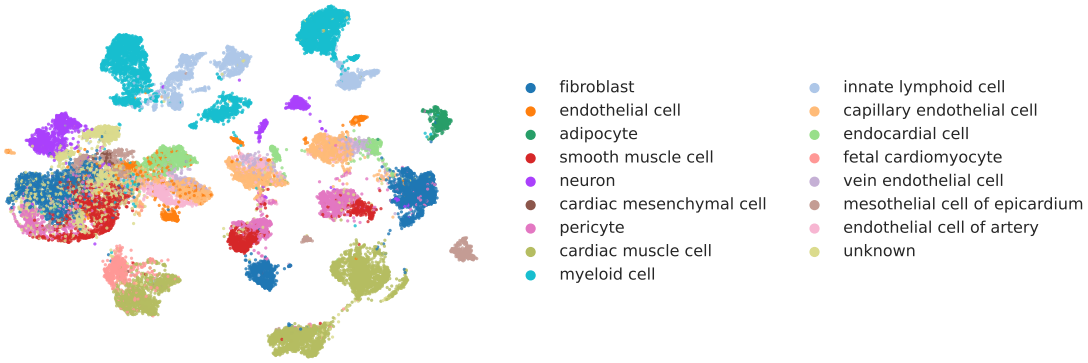

Figure 4: UMAP visualization of the heart dataset, where each point represents a single cell colored by its cell type.

3.1.2 Data Preprocessing

We designed the train, validation, and test splits for each dataset so that the evaluation reflects true generalization and avoids any form of information leakage between related samples. Before splitting, all datasets were processed with standard scRNA-seq quality-control steps, including removing cells with fewer than 200 detected genes and genes expressed in fewer than three cells.

For the Hematopoietic Niche dataset, which includes multiple donors and measurements taken at different timepoints, we performed splitting at the donor level. To maintain the biological diversity present in the data, we created a combined label that captures both the cell type and timepoint of each sample. We then identified a representative label for each donor and used it to guide stratified sampling. Donors were assigned to the training, validation, or test groups in a way that preserves the distribution of cell types and timepoints across the three sets.

For the PBMC68K dataset, donor information is not available, so we carried out stratified sampling directly at the level of individual cells. Cell type was used as the stratification label, allowing us to create a balanced split of eighty percent for training, ten percent for validation, and ten percent for testing while keeping all immune cell populations well represented.

For the Heart dataset, splitting was again performed at the donor level to avoid mixing samples from the same individual across different sets. We used each cell's assigned cell type to create stratification labels and selected a representative label for each donor. Using these labels, donors were distributed into training, validation, and test sets using stratified sampling, ensuring that each donor appears in only one set and that the biological composition remains consistent across the three groups.

After assigning all cells to their respective partitions, we saved the raw `.h5ad` files. This step was carried out using a multi-threaded processing pipeline with forty workers to handle the large datasets efficiently. To support all downstream evaluations, we created five completely independent versions of the train, validation, and test splits, each generated using a different random seed. These five splits were used throughout the experiments to provide reliable averages and confidence intervals.

3.2 Foundation Models

In this work, we evaluate four state-of-the-art foundation models designed for single-cell transcriptomics. scGPT, scFoundation, Geneformer, and scBERT. Although these models share the broad goal of learning generalizable representations from large-scale single-cell data, each introduces unique architectural ideas and pretraining strategies.

scGPT [31] is a generative transformer model pretrained on more than 33 million human single cells collected from the CELLxGENE database[3]. Inspired by large language models, scGPT treats genes as tokens and leverages masked generative pretraining with specially designed attention masks. This enables the model to simultaneously learn gene and cell embeddings that generalize across tissues, modalities, and experimental batches. scGPT achieves strong performance on downstream tasks such as cell-type classification, multi-batch integration, perturbation prediction, and gene regulatory network inference.

scFoundation [1] is a multimodal foundation model built on a large corpus of single-cell multi-omics profiles. It extends the transformer framework to incorporate gene expression, chromatin accessibility, and protein features, enabling rich cross-modality representations. By integrating multiple biological signals, scFoundation provides a strong backbone for downstream analyses such as multi-omic integration and regulatory feature interpretation, offering improved robustness in sparse or noisy settings.

Geneformer [30] is an attention-based transformer pretrained on 29.9 million human single-cell transcriptomes (Genecorpus-30M). It uses a rank-based, self-supervised masked modeling objective to learn context-aware gene and cell embeddings, focusing on capturing gene–gene interactions and network topology. Geneformer is particularly distinguished by its ability to infer regulatory relationships and perform context-specific tasks in settings with limited data, such as dosage sensitivity prediction, GRN inference, and patient-specific disease modeling.

scBERT [105] adapts the BERT architecture to single-cell data through a masked gene modeling objective on millions of scRNA-seq profiles. By representing genes as tokens and training the model to recover masked expression patterns, scBERT learns biologically meaningful embeddings that support tasks such as cell-type classification, dataset integration, and perturbation prediction. Its encoder-based design allows efficient fine-tuning with small labeled datasets and provides stable representations even in highly imbalanced or heterogeneous data.

Together, these four foundation models represent the current leading efforts to build general-purpose pretrained backbones for large-scale single-cell analysis. Table 1 provides a structured comparison of these four foundation models, summarizing their pretraining data, core architecture, and primary applications.

[3] https://cellxgene.cziscience.com/

Table 1: Summary of foundation models evaluated in this study.

Model	Pretraining Data Size	Architecture	Key Capabilities
scGPT[31]	∼33M human cells	transformer with masked attention	Cell-type annotation, batch correction, multi-omic integration, perturbation prediction, GRN inference
scFoundation[1]	Large multi-omics atlas (RNA, ATAC, proteins)	Encoder-decoder transformer backbone	Cross-modality integration, representation learning, scalable embeddings for sparse data
Geneformer[160]	29.9M human scRNA-seq (Genecorpus-30M)	Rank-based transformer with self-supervised masking	Context-aware network modeling, dosage sensitivity prediction, disease modeling
scBERT[105]	Millions of scRNA-seq profiles	BERT-style bidirectional encoder with masked gene modeling	Cell-type prediction, dataset integration, perturbation prediction, robust embeddings

3.3 Foundation Models Evaluation

In this section, we investigate how large single-cell foundation models behave when transferred to new datasets that were not seen during pretraining. Our methodology is structured into two main objectives. In this section, we evaluate four off-the-shelf foundation models; scGPT [31], scFoundation[1], Geneformer[30], and scBERT[105] across three diverse scRNA-seq datasets. This allows us to assess the generalization capacity of pretrained models when applied to unseen data and to examine how well their default fine-tuning pipelines adapt to different biological contexts.

3.3.1 Fine-Tuning Pipeline

All four foundation models are fine-tuned following their respective authors' recommended hyperparameters and training configurations. We treat fine-tuning as a transfer-learning adaptation step in which each model receives as input the raw gene expression matrix and optional metadata (depending on model design).

During fine-tuning, the transformer-based models update only a subset of layers or embedding blocks, depending on the model's guidelines. For example, Geneformer is typically fine-tuned using a lightweight classification head with frozen encoder blocks, whereas scGPT fine-tunes more of the transformer stack for downstream supervised tasks. scBERT and scFoundation incorporate additional architectural constraints (such as masked modeling heads or contrastive objectives) that are preserved during adaptation.

Across all models, the downstream task is formulated as a supervised multi-class classification problem for

cell-type prediction. The output is a probability distribution over annotated cell types, and model performance is evaluated using F1-score.

Evaluation Metric (F1-score): For all classification experiments, we report the F1-score as the primary evaluation metric. F1-score provides a balanced measure of precision and recall, making it particularly suitable for scRNA-seq datasets where class distributions are often highly imbalanced and rare cell populations may be overshadowed by abundant ones.

For each class c, precision and recall are defined as:

$$\text{Precision}_c = \frac{\text{TP}_c}{\text{TP}_c + \text{FP}_c}, \qquad \text{Recall}_c = \frac{\text{TP}_c}{\text{TP}_c + \text{FN}_c}, \tag{1}$$

The F1–score for class c is the harmonic mean of precision and recall:

$$\text{F1}_c = \frac{2 \cdot \text{Precision}_c \cdot \text{Recall}_c}{\text{Precision}_c + \text{Recall}_c}. \tag{2}$$

Objective: During finetuning, we use a multiclass crossentropy objective suitable for cell type classification. Given the predicted class probabilities $\hat{y}_{b,c}$ for cell b and class c, and the onehot ground truth label $y_{b,c}$, the loss is:

$$\mathcal{L} = -\frac{1}{B} \sum_{b=1}^{B} \sum_{c=1}^{C} y_{b,c} \log \hat{y}_{b,c} \tag{3}$$

The first term is the standard crossentropy used for multiclass classification, encouraging the model to assign high probability to the correct cell type.

3.3.2 Dataset-Specific Evaluation

Although the same fine-tuning pipeline is applied to all datasets, our depth of analysis varies:

- **PBMC68K** and **Heart Dataset:** These datasets are used primarily for benchmarking. For each model, we report standard classification metrics and compare performance across models to assess generalization and robustness. No additional biological interpretation or error analysis is performed on these datasets.
- **AML (Hematopoietic Niche) Dataset:** While AML is treated identically during the fine-tuning stage, we perform an extended analysis on this dataset due to its biological complexity, rich cell-type diversity, and relevance to disease-state modeling. After fine-tuning, we examine the per-class behavior of each model, investigate representation quality, and study feature importance patterns. This deeper analysis allows us to understand why some models generalize better than others in disease settings.

3.3.3 Summary

Overall, this methodological framework enables a controlled, fair comparison of large pretrained single-cell foundation models across multiple datasets. Consistent fine-tuning procedures ensure that observed performance differences arise from the models themselves rather than from discrepancies in training strategy. The AML dataset subsequently serves as a case study to probe the internal behavior of each model in greater detail, while the other datasets provide broad benchmarking across diverse biological conditions.

3.4 Multinomial Attention Masking (MAM) Algorithm

In the second objective of this study, we move beyond evaluating existing foundation models and introduce our own approach to improve how these models learn from sparse and high dimensional scRNA seq data. One important observation is that biological relevance is not determined simply by whether a gene is zero or highly expressed. Genes with zero expression can still be informative in specific cellular contexts, while genes that are highly and consistently expressed across all cells often provide limited insight into cell identity or state. Motivated by this, we replace conventional random masking with a more biologically grounded strategy. We propose the Multinomial Attention Masking algorithm, which leverages learned attention patterns to identify gene positions that are informative in context during self supervised training. We also introduce a cross attention based context injection mechanism that feeds global latent information back into the backbone model, helping it better understand which parts of the input matter for reconstruction. This section focuses on improving the pretraining process itself, making it more adaptive, interpretable, and better aligned with the structure of single cell data, while remaining compatible with a wide range of foundation model architectures.

3.4.1 Backbone Foundation Model

Our goal was to show that our approach is compatible with encoder, decoder, and encoder-decoder FMs. Due to the heavy computation cost of pretraining FMs, without loss of generality, we used scFoundation [1] as our backbone. To the best of our knowledge, scFoundation is the only open-source FM that uses both an encoder and a decoder. We note that among other FMs, scBERT [105] and Geneformer [30] are encoder-only models, scGPT [31] is a encoder-only model that does not provide the pretraining code. Therefore, computationally pretraining scBERT and scGPT and testing different scenarios of finetuning and linear probing on different datasets was infeasible with our resources. Additionally, we ensured that the datasets used in our experiments do not overlap with those in the pretraining corpus of the scFoundation model [1].

3.4.2 Latent Cross-Attention Module

We consider an input dataset consisting of gene expression profiles from single cells. Each cell is treated as a separate data instance, represented by a sequence of gene tokens. For a given cell, the input sequence is a vector of gene expression values ordered by gene ID.

We introduce $Z \in \mathbb{R}^{N \times D_z}$, a trainable set of N latent vectors, each of embedding dimension D_z. These latent vectors attend over input token sequences $X \in B \times L$, where B is the batch size (number of cells) and L is the sequence length (number of genes). Then, we project X to $X_{proj} \in B \times L \times D_z$ using a simple, trainable linear projection layer to make the dimensions aligned with latents. The ultimate goal is to extract a global summary that guides downstream masking decisions as depicted in Fig. 5. Formally, let:

$$Z = \left[z_1, z_2, \ldots, z_N\right]^\top \in \mathbb{R}^{N \times D_z}, \quad z_n \in \mathbb{R}^{D_z} \tag{4}$$

We apply multi-head cross-attention with H heads, using Z as the set of queries and X_{proj} as keys and values.

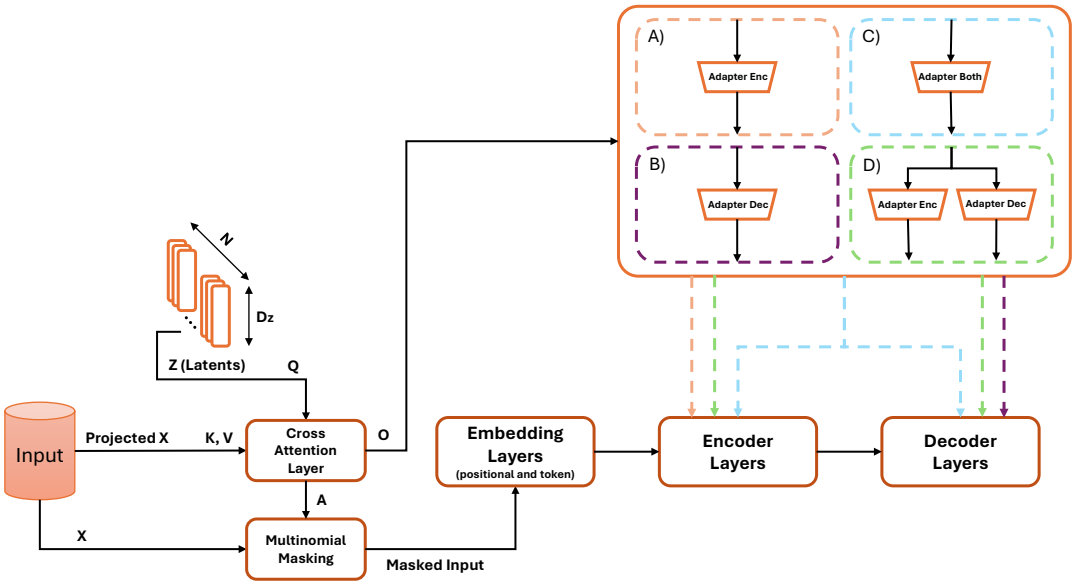

Figure 5: Proposed attention-driven masking and context-injection workflow. Learned latent vectors attend to token embeddings via cross-attention (Q, K, V) to produce attention weights A and pooled context O. A multinomial sampling of A yields the mask positions. The masked sequence is passed through encoder and decoder layers, with O injected according to four strategies: (A) encoder-only, (B) decoder-only, (C) both with a shared adapter, or (D) both with separate adapters.

To match the shape requirements of standard multi-head attention implementations, we reshape:

$$Q = Z \in \mathbb{R}^{N\times B\times D_z}, \quad K = V = X_{proj} \in \mathbb{R}^{L\times B\times D_z}. \tag{5}$$

The cross-attention then produces:

$$\big(O, A\big) = \text{CrossAttn}(Q, K, V), \tag{6}$$

where $O \in \mathbb{R}^{N\times B\times D_z}$ (attended latent outputs) and $A \in \mathbb{R}^{B\times H\times N\times L}$ (attention weights per head).

3.4.3 Multinomial Attention Masking (MAM)

To find the important tokens for masking, we use per-head attention weights. Given the raw attention weights, A, we generate a masking pattern via a sequence of steps. First, we aggregate across the H cross-attention heads by summing:

$$S_{b,n,t} = \sum_{h=1}^{H} A_{b,h,n,l}, \quad \textit{where} \quad S \in \mathbb{R}^{B\times N\times L}. \tag{7}$$

This collapses the per-head weights into a single attention score for each latent n and token position l. Second, to focus on a subset of latent vectors, we randomly select $R \subset \{1,\ldots,N\}$. Only the R latents are

used to drive masking, reducing noise from less-informative latents. As noted in SMA[159], attention maps often exhibit heavy overlap. Many queries focus on highly similar or correlated token groups. This makes it difficult to construct diverse or informative masking patterns when using all attention heads or latent vectors simultaneously, resulting in overly concentrated, redundant masking. We then collapse these selected latents' scores into token-level importance by summing across selected latents.

$$s_{b,l} = \sum_{n \in R} S_{b,n,l}, \quad s \in \mathbb{R}^{B \times L} \tag{8}$$

Here, matrix s quantifies each token's overall attention across the chosen latents. To convert these scores into a valid sampling distribution, we add a small constant $\varepsilon > 0$ (division-by-zero avoidance) and normalize:

$$p_{b,l} = \frac{s_{b,l} + \varepsilon}{\sum_{l'=1}^{L} \left(s_{b,l'} + \varepsilon\right)}, \quad \textit{where} \quad \sum_{t=1}^{L} p_{b,l} = 1. \tag{9}$$

Finally, for each cell b, we sample (multinomial sampling) $k = \lfloor \rho \times L \rfloor$ distinct token indices from the probability distribution $p_{b,1}, \ldots, p_{b,L}$. We then mask these k positions, leaving all remaining tokens visible to both the encoder and the decoder.

3.4.4 Cross-Attention Context Injection Strategies

To enable the latent vectors to learn which token features are most informative and exploring masking information throughout the modules, we inject the cross-attention outputs from Eq. 6 back into the backbone network. First, we pool O over the N latents:

$$o_b \;=\; \frac{1}{N} \sum_{n=1}^{N} O_{n,b,:} \;\in\; \mathbb{R}^{D_z}, \quad b = 1, \ldots, B, \tag{10}$$

The context vector o_b is then injected into the encoder and/or decoder streams to guide reconstruction. We explore three injection schemes, each preceded by a single-layer linear adapter that adapts the scale and distribution of the context, and project o_b back to the token embedding dimension D_e:

$$\begin{aligned} \text{adapter}(o_b) \;&=\; W_{\text{adapt}}\, o_b \;+\; b_{\text{adapt}}, \\ W_{\text{adapt}} \in \mathbb{R}^{D_e \times D_z}; \quad & b_{\text{adapt}} \in \mathbb{R}^{D_e}. \end{aligned} \tag{11}$$

Here, W_{adapt} and b_{adapt} are learnable parameters. There are four different ways (Fig. 5) to use the adapter(o_b) output:

Encoder-only: In this case, we let the encoder see the cross-attention and help the latents to be trained properly. Our intuition is that the encoder helps the latents to learn the deep contexts of the data so they can understand which parts of the input are more important.

Decoder-only: We add the attention output to the decoder. In this case, decoder helps latents to understand the reconstruction process, and which parts of the data are more challenging to reconstruct.

Encoder and Decoder: We broadcast the same adapter-transformed context into both the encoder and decoder. This "merged" approach gives latents a holistic view—first to identify important inputs, then to assist reconstruction—while keeping the number of extra parameters minimal by using a single adapter.

Encoder and Decoder with Separate Learnable Adapters: We use two independent single-layer adapters for encoder injection and one for decoder injection, so that each can learn a specialized transformation of the latent summary. The encoder adapter can emphasize features that aid in representation learning, while the decoder adapter can highlight aspects that improve reconstruction, without forcing a one-size-fits-all mapping.

3.4.5 Self-supervised Pretraining

In this step, the network learns to reconstruct the masked gene expression values using the proposed attention-driven masking. We used pretrained weights of our backbone because training from scratch requires extensive computing resources. We experimented: 1) masking all suggested tokens, and 2) via the "80/10/10" scheme (80% mask token, 10% random gene, 10% unchanged) on all experiments, to see which one is more effective in our case.

We trained the backbone separately on each of our three datasets. For all datasets, we apply a masking ratio of 30% during pretraining, same as our backbone model default setup. This ratio is used consistently across all methods, ensuring a fair comparison. Our attention-driven masking dynamically selects which positions to mask at each step, allowing the model to focus on informative input regions while maintaining sufficient context for learning. The random masking baseline uses the same 30% ratio and follows the standard strategy previously used with the backbone model.

Moreover, we explore four distinct context-injection strategies (As explained in section 3.4.4 and Figure 5). In scheme (A), the pooled cross-attention vector is added only to the encoder. Scheme (B) injects the same context solely into the decoder. Scheme (C) uses a single shared adapter to broadcast the transformed context into both the encoder and decoder. Finally, scheme (D) employs two separate single-layer adapters—one dedicated to the encoder and one to the decoder—allowing each to learn its own transformation of the latent summary. We conduct experiments to evaluate the effectiveness of each of these four variants.

Objective: Our self-supervised loss combines a mean-squared reconstruction term over the masked positions with an L_2 penalty on the cross-attention outputs O:

$$\mathcal{L} = \frac{1}{|\mathcal{M}|} \sum_{(b,t)\in\mathcal{M}} (\hat{x}_{b,t} - x_{b,t})^2 \; + \; \lambda_{\text{attn}} \|O\|_2^2, \qquad \lambda_{\text{attn}} = 10^{-5}. \tag{12}$$

The reconstruction term forces the decoder to recover masked gene expression values, while the L_2 regularization on the cross-attention outputs O prevents these activations from growing without bound (avoiding unstable or inflated latent representations), encourages smoother, less peaky attention distributions (promoting robust context aggregation across genes), and dampens dataset-specific idiosyncrasies (reducing overfitting). Together, these effects keep the model's internal representations well-behaved and improve its ability to generalize to new cells or perturbations.

Hyperparameters: We train the parameters using Adam with two learning-rate groups: $\mathrm{LR}_{\mathrm{new}} = 10^{-4}$ for layers introduced by our method and $\mathrm{LR}_{\mathrm{backbone}} = 10^{-5}$ for the rest. Gradients are clipped to norm 1.0, accumulated over five steps (effective batch = 5), and subject to weight decay 5×10^{-4}. Early stopping is applied after five epochs without validation loss improvement (following the default pretraining hyperparameters in the backbone's original paper [1]).

3.4.6 Supervised Finetuning Setup

A lightweight classification head is added on top of the pretrained encoder and trained on downstream tasks. We focus on cell type classification, an important bioinformatics task supported by all evaluated datasets. For each dataset, we finetune the model to predict annotated cell types by appending a two-layer MLP on top of the pooled encoder output. During finetuning and inference, the decoder is completely removed, and only the encoder together with the classification head is used.

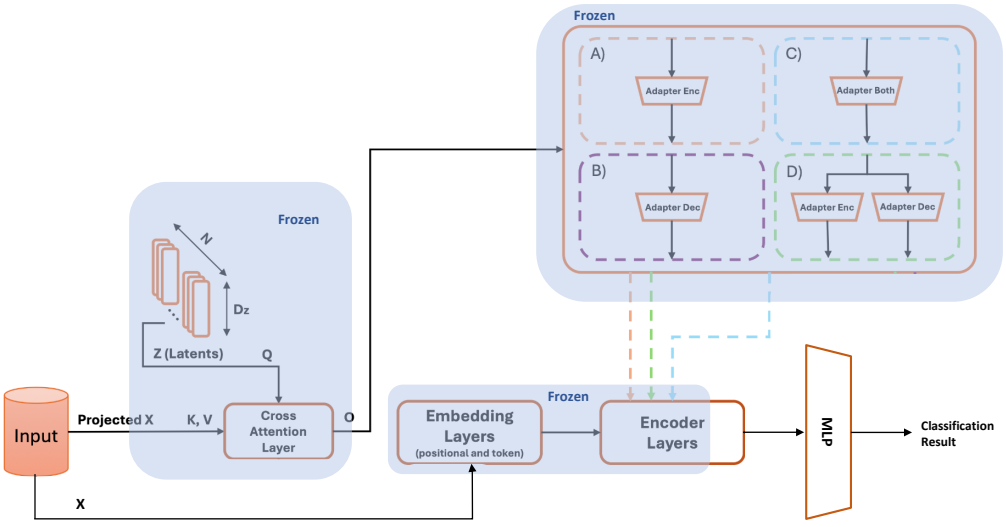

Figure 6: Supervised finetuning setup for downstream classification. During finetuning, latent tokens, cross-attention layers, and all adapters remain frozen. The MLP outputs a probability distribution over cell types for evaluation using the F1-score.

We freeze all latent tokens, cross-attention weights, and all adapter layers during finetuning, as these components are introduced solely for self-supervised pretraining. While the decoder is removed, any cross-attention

outputs that were originally injected into the encoder during pretraining are still passed to the encoder in finetuning if they were part of the encoder pathway. In contrast, cross-attention outputs that were exclusively used by the decoder naturally become inactive once the decoder is removed.

We evaluate both partial finetuning and linear probing schemes (on top of the classification head) to assess the usefulness of the pretrained encoder representations. Full finetuning is omitted due to the extensive computational resources required by the model. In partial finetuning, the trainable parameters include the token embeddings, positional embeddings, the last transformer block of the encoder, and the classification head, while all earlier encoder layers remain frozen.

For linear probing, only the classification head is trainable. All encoder layers, token embeddings, and positional embeddings are frozen. In both finetuning schemes, all masking mechanisms and cross-attention modules are frozen. These components are designed to enhance robustness during pretraining, and allowing them to update during finetuning would effectively introduce additional trainable parameters beyond the classification head, which is not our goal. The model outputs a probability distribution over annotated cell types, and performance is evaluated using the F1-score.

Evaluation Metric (F1-score): For all classification experiments, we report the F1-score as our primary evaluation metric. The F1-score balances precision and recall, making it especially appropriate for scRNA-seq tasks where class distributions are highly imbalanced and rare cell types can easily be overwhelmed by more abundant populations.

For each class c, precision and recall are defined as:

$$\text{Precision}_c = \frac{\text{TP}_c}{\text{TP}_c + \text{FP}_c}, \qquad \text{Recall}_c = \frac{\text{TP}_c}{\text{TP}_c + \text{FN}_c}, \tag{13}$$

The F1–score for class c is the harmonic mean of precision and recall:

$$\text{F1}_c = \frac{2 \cdot \text{Precision}_c \cdot \text{Recall}_c}{\text{Precision}_c + \text{Recall}_c}. \tag{14}$$

Objective: During finetuning, we use a multiclass crossentropy objective suitable for cell type classification. Given the predicted class probabilities $\hat{y}_{b,c}$ for cell b and class c, and the onehot ground truth label $y_{b,c}$, the loss is:

$$\mathcal{L} = -\frac{1}{B} \sum_{b=1}^{B} \sum_{c=1}^{C} y_{b,c} \log \hat{y}_{b,c} \; + \; \lambda_{\text{attn}} \|O\|_2^2, \qquad \lambda_{\text{attn}} = 10^{-5}. \tag{15}$$

The first term is the standard crossentropy used for multiclass classification, encouraging the model to assign high probability to the correct cell type. The second term applies an L_2 penalty on the crossattention outputs O, preventing excessively large activations and promoting stable, smooth attention patterns during finetuning.

3.4.7 Biological Evaluation

Finally, to show that our approach (MAM) masks the most important genes, we tried two biological experiments: First, we analyzed the relation between the average of expression values and the number of times that each gene is masked. This can help us understand the pattern of masked genes. Second, we attempted Gene Ontology (GO) enrichment [247], which identifies the biological processes of a set of genes. Using GO enrichment, we can determine if the masked gene is informative or not.

3.4.8 Summary

Our method introduces Multinomial Attention Masking (MAM), a biologically informed masking strategy that replaces purely random masking during fine-tuning. We use scFoundation as the main backbone due to its open encoder–decoder architecture, but the method is compatible with encoder-only (scBERT, Geneformer) and decoder-only (scGPT) models. A set of trainable latent vectors performs cross-attention over gene expression tokens, producing attention maps that highlight informative genes. These scores are aggregated and transformed into a multinomial distribution, which we sample from to choose the mask positions. This makes the masking step adaptive, guided directly by the model's own learned understanding of gene importance.

To further support the learning process, we inject a summary of the latent attention outputs back into the backbone model. This context vector is added to the encoder, decoder, or both, helping the model emphasize meaningful features during both representation learning and reconstruction. Different injection variants allow us to control how much global information flows through the network. Together, attention-driven masking and context injection create a more interpretable and effective fine-tuning framework that adapts well across different datasets and foundation model architectures.

4 Experiments

4.1 Foundation Model Evaluation

4.1.1 Training Hyper-parameters

In this section, we summarize the hyper-parameter configurations used to fine-tune the four foundation models evaluated in our study. Although each model differs substantially in architecture and pretraining strategy, we aligned their fine-tuning setups as closely as possible to enable a fair comparison. Table 2 reports the final hyper-parameters adopted for scBERT[105], Geneformer[160], scGPT[31], and scFoundation[1].

Table 2: Training hyper-parameters used for fine-tuning each foundation model.

Hyper-parameter	**scBERT**	**Geneformer**	**scGPT**	**scFoundation**
Epochs	10	10	10	10
Batch size	8	16	8	4
Learning rate	1×10^{-4}	1×10^{-4}	1×10^{-4}	1×10^{-4}
Gradient accumulation	60	—	—	5
Early stopping patience	5	5	5	5

All models were fine-tuned for ten epochs, which we found to provide stable convergence across datasets without signs of overfitting. The learning rate was fixed at 1×10^{-4} for every model, following common practice in transformer-based architectures and ensuring a consistent optimization regime across experiments. Batch sizes differed slightly due to memory constraints intrinsic to each model: Geneformer, which operates on compact rank-based gene encodings, could support a larger batch size (16), whereas scFoundation required a smaller batch size (4) to accommodate its larger per-cell representation. scBERT and scGPT were trained with a batch size of 8.

Two models rely on gradient accumulation to simulate larger effective batch sizes. scBERT required substantial accumulation (60 steps) because of its high memory footprint, whereas scFoundation used a smaller accumulation factor of 5. Geneformer and scGPT do not use gradient accumulation during fine-tuning. To prevent overfitting and ensure consistent convergence behavior across datasets, all models were trained with an early-stopping patience of five validation checks. This setting offered a stable trade-off between training efficiency and robustness.

4.1.2 Computing Resources

Table 7 provides an overview of the computing setup and the approximate runtimes for the foundation model experiments. All runs were carried out on a standardized hardware configuration of four NVIDIA A40 GPUs to maintain consistency across datasets and architectures. For each dataset–model pair, the table lists the estimated end-to-end time required for fine-tuning and evaluation. The final row summarizes the total runtime across all experiments, offering a clear sense of the overall computational effort involved.

Table 3: Compute resources and end-to-end runtime for all foundation model evaluations. Each experiment used 4 × NVIDIA A40 GPUs (48 GB each). Runtime estimates reflect fine-tuning and evaluation only.

Dataset	Model	Approx. Runtime
Hematopoietic Niche	scBERT	∼ 1 week
	Geneformer	∼ 1 week
	scGPT	∼ 1 week
	scFoundation	∼ 1 week
PBMC 68K	scBERT	∼ 1 week
	Geneformer	∼ 1 week
	scGPT	∼ 1 week
	scFoundation	∼ 1 week
Heart	scBERT	∼ 1 week
	Geneformer	∼ 1 week
	scGPT	∼ 1 week
	scFoundation	∼ 1 week
Total	—	**∼ 12 weeks**

4.1.3 Overall Model Performance Across Datasets

Understanding how foundation models generalize across diverse biological contexts is essential for evaluating their robustness and practical applicability. To this end, we systematically compared the performance of four state-of-the-art models (Geneformer[160], scBERT[105], scFoundation[1], and scGPT[31]) across three benchmark single-cell datasets that vary widely in tissue origin, cellular complexity, and annotation difficulty.

This section provides an integrated view of these comparisons, highlighting where models perform consistently well, where they diverge, and how dataset-specific characteristics shape their strengths and limitations. As summarized in Table 4 and illustrated by the performance visualizations in Figures 7–9, the four foundation models exhibit distinct generalization patterns across the Hematopoietic Niche, PBMC 68K, and Heart datasets.

Hematopoietic Niche dataset: Across the Hematopoietic Niche dataset, all four foundation models achieved moderate to strong performance, but with clear separation between the highest- and lowest-performing approaches. Geneformer attained the strongest overall results with an F1-score of 82.3%, followed closely by scFoundation (80.8%) and scGPT (80.5%). In contrast, scBERT lagged substantially, reaching only 74.2%. These differences suggest that models leveraging more extensive pretraining on single-cell dataset, such as Geneformer and scFoundation, generalize more effectively to the complex cell-state landscape characteristic of hematopoietic tissue. The relatively lower performance of scBERT reflects challenges in capturing lineage hierarchies and subtle transcriptional gradients present in this dataset.

PBMC 68K dataset: A different pattern emerged in the PBMC 68K dataset, where all models performed slightly higher overall but still exhibited marked variation. Here, scGPT achieved the highest F1-score of 84.2%, with Geneformer close behind at 82.9%. scBERT and scFoundation followed with F1 scores of

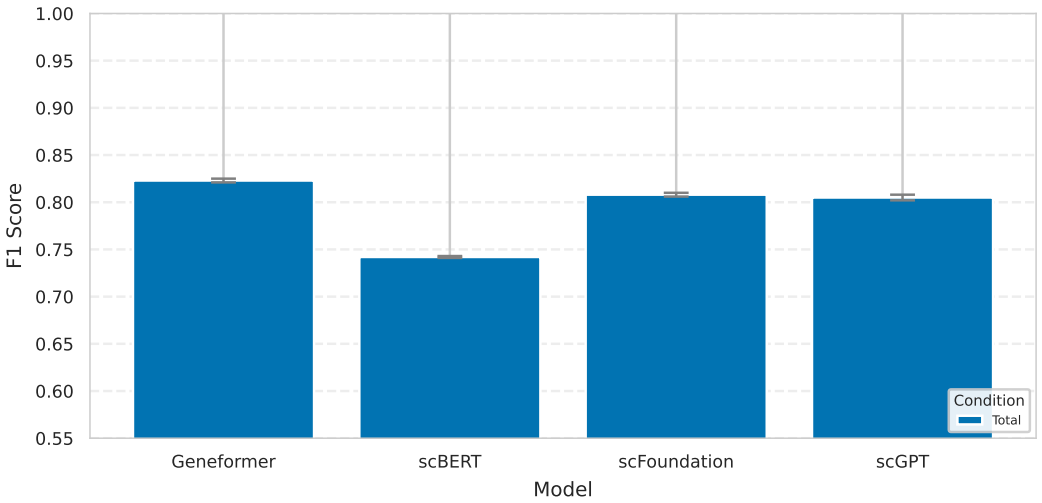

Figure 7: Performance of four foundation models (Geneformer, scBERT, scFoundation, scGPT) on Hematopoietic Niche dataset.

79.3% and 77.6%, respectively. The relative compression of performance gaps in this dataset suggests that distinguishing major immune populations is a more accessible task for all models, though scGPT in particular appears to leverage pretraining dynamics that confer an advantage when resolving fine-grained immune cell identities.

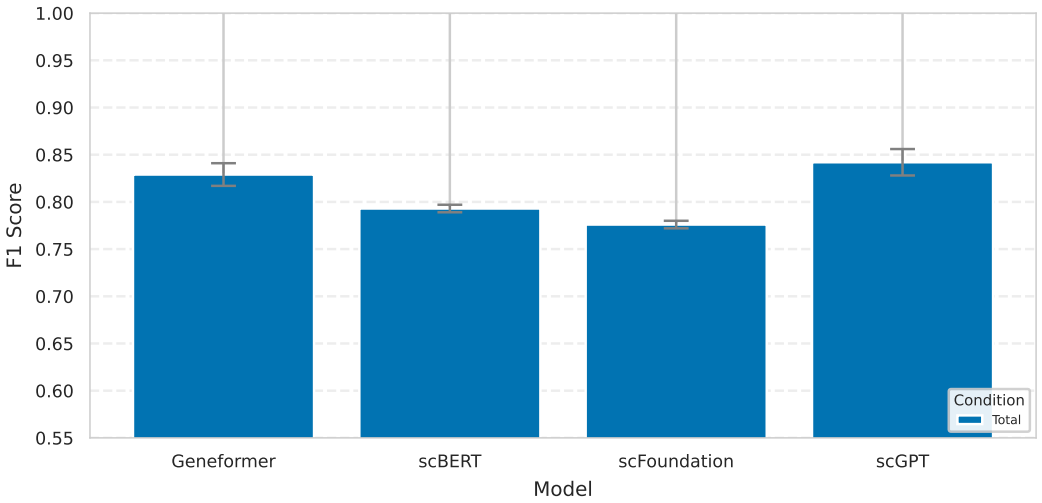

Figure 8: Performance of four foundation models (Geneformer, scBERT, scFoundation, scGPT) on PBMC 68K dataset.

Heart dataset: In the Heart dataset, the models displayed their highest separation in overall performance. scBERT and scFoundation were the strongest performers, reaching F1-score values of 91.7% and 91.6%, respectively. Geneformer performed more modestly at 79.2%, while scGPT exhibited an unexpectedly large drop, with an F1-score of only 83%.

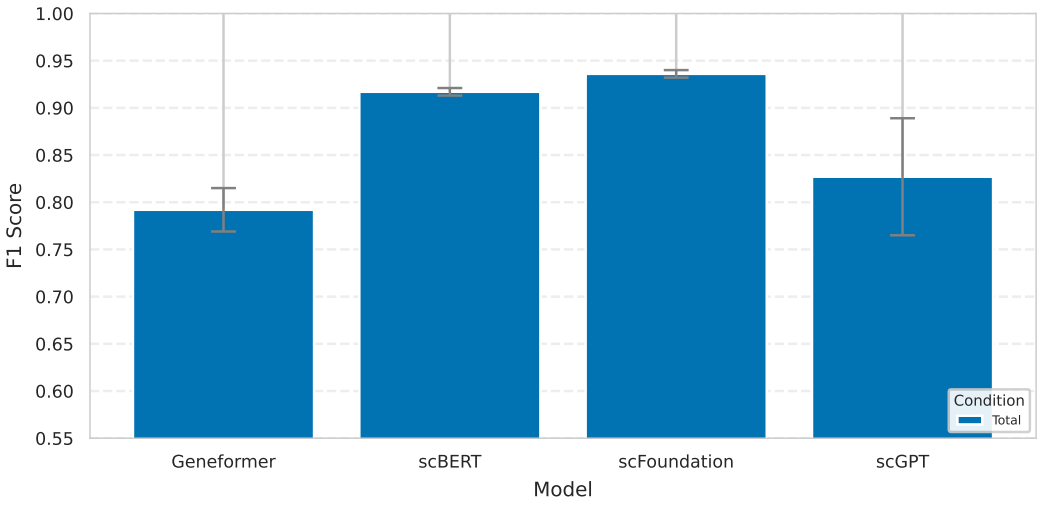

Figure 9: Performance of four foundation models (Geneformer, scBERT, scFoundation, scGPT) on Heart dataset.

The strong results for scBERT and scFoundation indicate that these architectures are particularly well-suited for datasets dominated by highly specialized cell types with structured transcriptional programs. Conversely, the degradation observed for scGPT suggests challenges in modeling tissue-specific signatures when they diverge substantially from patterns represented in its pretraining data. Together, these results underscore that model generalization is strongly dataset-dependent, with each biological context amplifying distinct strengths and limitations of current single-cell foundation models.

Table 4: F1-score performance of four foundation models across three datasets. Results are reported as the mean ± confidence interval from 5-fold cross-validation on the test set, with each fold held out once.

Model	Hematopoietic Niche	PBMC 68K	Heart
Geneformer	82.3 ± 0.002	82.9 ± 0.012	79.2 ± 0.022
scBERT	74.2 ± 0.001	79.3 ± 0.004	91.7 ± 0.004
scFoundation	80.8 ± 0.002	77.6 ± 0.004	91.6 ± 0.004
scGPT	80.5 ± 0.003	84.2 ± 0.014	83.0 ± 0.006

4.1.4 In-Depth Analysis on the Hematopoietic Niche Dataset

A closer look at model performance within the Hematopoietic Niche dataset uncovers patterns that are not apparent from overall metrics alone. Unlike the PBMC 68K and Heart datasets, which do not include clinical timepoint information and therefore cannot support disease-state analysis, the Hematopoietic Niche dataset offers detailed annotations spanning healthy, diagnostic, post-treatment, and relapse conditions.

This richer metadata makes it an ideal setting for examining how foundation models respond to clinically meaningful variation. Because pretraining datasets are often heavily skewed toward healthy cells, models may generalize unevenly across disease states or struggle with rare cellular populations. To better understand these effects, we perform a stratified evaluation across both clinical timepoints and individual cell types,

allowing us to expose systematic failures that would be invisible in aggregate scores. This fine-grained analysis highlights not only where current foundation models perform well, but also where they fall short, and why certain biologically and clinically important subgroups remain challenging despite strong overall performance.

Performance Across Clinical Timepoints: To evaluate how foundation models respond to variation in clinical state, we stratified performance by health and disease conditions. When averaging across all four models (Geneformer, scBERT, scFoundation, and scGPT [160, 105, 1, 31]), we observed consistently higher F1 scores for Healthy donors compared to AML samples (Fig. 10). Within the AML cohort, relapse cases were consistently the most challenging, exhibiting a more pronounced decline in model performance than primary AML samples.

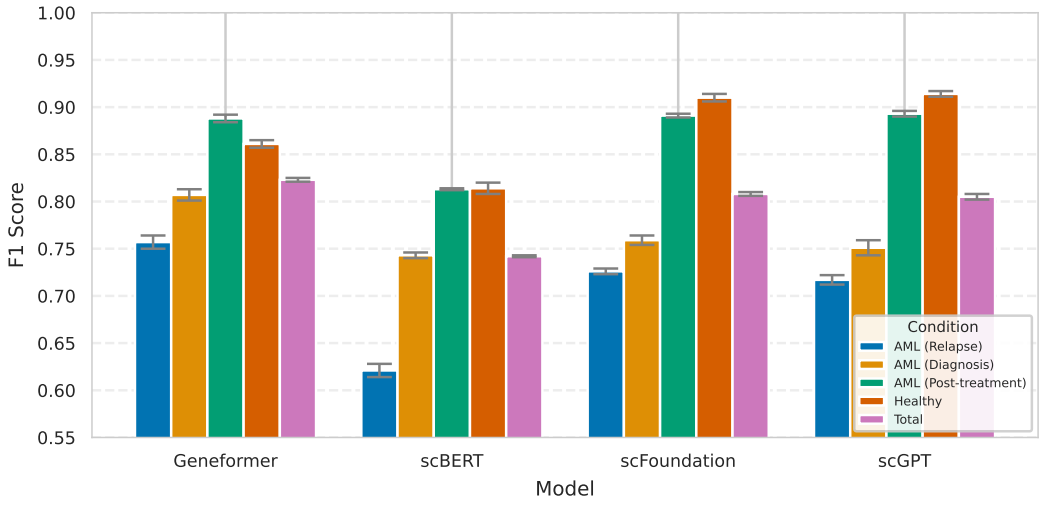

Figure 10: Performance of four foundation models (Geneformer, scBERT, scFoundation, scGPT) stratified by clinical timepoint. While models perform well on Healthy samples, performance drops in AML, with the largest decline observed in relapse cases. This pattern highlights a systematic gap: foundation models pretrained on mostly healthy single-cell data generalize poorly to clinically challenging disease states, even though our evaluation dataset was evenly balanced across all timepoints.

For example, scGPT maintained F1 scores above 90% in Healthy donors but dropped below 75% in relapse samples, with scBERT and scFoundation showing similar reductions. These trends indicate that model generalization is strongly condition-dependent, and that the transcriptional complexity associated with relapse introduces substantial difficulty for current foundation models. Collectively, these findings highlight the need for more robust representations capable of capturing the molecular heterogeneity characteristic of disease progression.

Performance Across Cell Types: To examine how foundation models perform across distinct cellular populations, we evaluated their predictive accuracy on each cell type individually. As shown in Fig. 11, all models exhibit substantial difficulty identifying several clinically relevant populations, including CLP and

Pre.B cells—cell types whose depletion is closely associated with the immune dysfunction characteristic of AML patients [248].

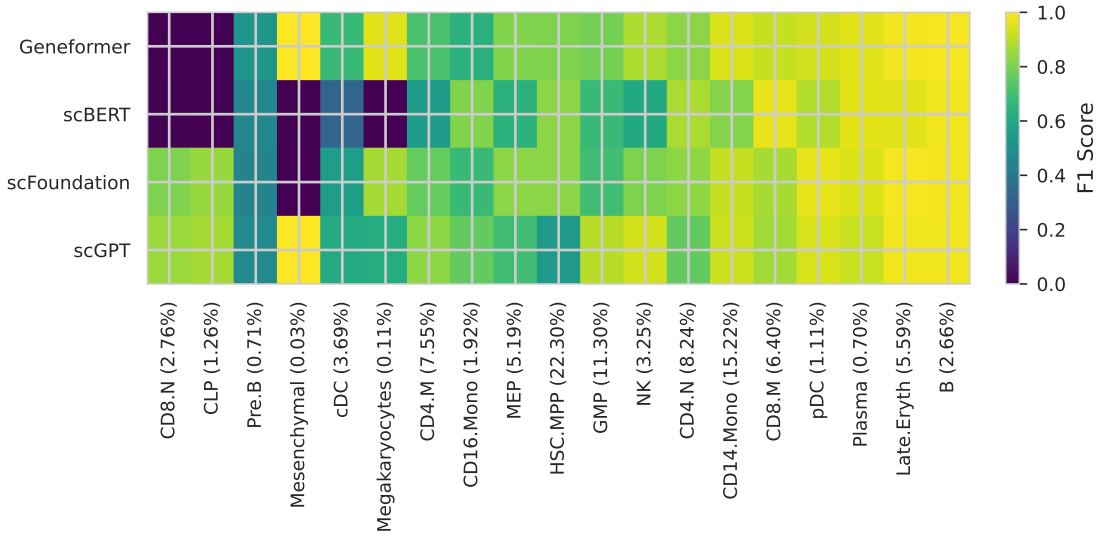

Figure 11: Heatmap of F1 scores by cell type across four foundation models. Performance is uneven: common and well-represented populations such as B cells show strong accuracy, while others like CLP, Pre.B exhibit markedly lower scores. In addition, some cell types are totally mistaken by foundation models, resulting in zero in their F1 Score.

In contrast, performance on more abundant and broadly defined populations, such as general B cells, remains consistently high across models. This disparity highlights how under-represented or transcriptionally subtle cell types introduce systematic vulnerabilities, ultimately constraining the clinical reliability of current foundation-model predictions.

4.1.5 Summary

Section 1 established the foundation for our benchmarking study by providing a controlled and comprehensive evaluation of four widely used single-cell foundation models. Our goal in this section was to fine-tune each model under comparable conditions, quantify how well they generalize across biologically diverse datasets, and build a deeper understanding of the factors that shape their performance.

Evaluating performance across the Hematopoietic Niche, PBMC 68K, and Heart datasets revealed that no model performs uniformly across all biological contexts. Instead, each dataset brought forward different strengths and weaknesses. Geneformer, scFoundation, and scGPT performed strongly on the Hematopoietic Niche and PBMC 68K datasets, while scBERT and scFoundation achieved the highest scores on the Heart dataset, where cell types are more specialized and transcriptionally distinct. These results, summarized in Table 4, illustrate how differences in tissue composition and annotation complexity can shift the relative advantages of each model.

The Hematopoietic Niche dataset provided an opportunity for even deeper analysis. Unlike the PBMC 68K

and Heart datasets, it includes detailed clinical timepoint annotations, enabling us to examine how models behave across healthy, diagnostic, post-treatment, and relapse conditions. This richer metadata revealed systematic performance drops in all AML states, with relapse cells posing the greatest challenge. A complementary analysis at the cell-type level showed similarly uneven performance: common populations such as B cells were classified reliably, whereas rare or transcriptionally subtle types, including CLP and Pre.B cells, remained difficult for all models. These findings point to the lingering influence of pretraining biases and the challenges posed by rare or underrepresented cell states.

Overall, our findings highlights both the promise and limitations of current single-cell foundation models. Although they achieve strong accuracy on aggregate, their performance varies substantially across datasets, disease conditions, and cell types.

4.2 Multinomial Attention Masking (MAM) Algorithm

In the second part of our experiments, we move beyond evaluating off-the-shelf foundation models and introduce our proposed Multinomial Attention Masking (MAM) algorithm. This objective focuses on improving the masked-autoencoding objective commonly used during pretraining of single-cell foundation models. Traditional random masking, used in models such as scGPT[31], scBERT[105], Geneformer[30], and scFoundation[1], treats all genes equally, despite the extreme sparsity and uneven biological relevance of scRNA-seq data.

As shown in our reference work, masking strategies in single cell models must carefully reflect the structure of gene expression data. Zero valued genes do not imply a lack of biological meaning, as their relevance is often context dependent and can represent meaningful cellular states. Likewise, high expression values alone do not determine importance. Genes that are consistently expressed with low variability across cells, even when highly expressed, are often less informative for learning discriminative representations. MAM naturally accounts for both properties by learning data driven masking patterns from the model, allowing gene importance to be inferred from contextual relationships rather than expression values or sparsity alone. In this section, we integrate MAM into the scFoundation cite scFoundation backbone and evaluate its impact across all datasets, demonstrating how attention guided masking and context injection improve model robustness, interpretability, and downstream performance.

4.2.1 Training Hyper-parameters

Before describing the individual hyperparameters used in our experiments, it is important to note that all configurations were selected through a combination of empirical hyperparameter tuning and alignment with the default recommendations provided by the scFoundation[1] backbone model. Since the backbone was originally optimized for large-scale scRNA-seq pretraining, we adopted similar parameter ranges to ensure stable training, and adapted them only when necessary to accommodate our additional MAM modules and the computational resources available. The final hyperparameters reported below represent the best balance between training stability, computational efficiency, and downstream performance across all datasets.

Self-supervised Pretraining: During the self-supervised pretraining stage, we configure the model to learn robust representations from raw gene expression profiles before introducing any labeled information. The latent space is defined by a latent embedding dimension of $D_z = 256$ and a total of $N = 128$ trainable latent vectors, which collectively form the basis for our multinomial masking decisions.

Cross-attention modules use 8 attention heads, enabling the latents to attend to different subsets of genes simultaneously. To stabilize optimization, gradients are clipped to a maximum norm of 1.0, preventing exploding gradients in long training runs. Training updates use two learning-rate groups: newly introduced layers (such as latent vectors, cross-attention, and adapters) are trained with a higher learning rate of 1×10^{-4}, while the backbone layers receive a smaller learning rate of 1×10^{-5} to avoid overwriting previously learned representations.

A weight decay of 5×10^{-4} is applied to all trainable parameters for regularization. The attention regularizer $\lambda_{\text{attn}} = 1 \times 10^{-5}$ penalizes overly sharp cross-attention activations. Because scRNA-seq data exceed GPU memory limits, the physical batch size is set to 1, and gradients are accumulated over 5 steps, resulting in an effective batch size of 5. Training uses early stopping with a patience of 5 epochs to prevent overfitting once the validation loss plateaus. Table 5 presents the hyperparameters we have used for self-supervised pretraining.

Table 5: Hyperparameters for self-supervised pretraining (MAM).

Parameter	**Value**	**Parameter**	**Value**
Latent dimension D_z	256	# latents N	128
Cross-attn heads	8	Gradient clipping	1.0
LR (new layers)	1×10^{-4}	LR (backbone)	1×10^{-5}
Weight decay	5×10^{-4}	λ_{attn}	1×10^{-5}
Batch size (physical)	1	Accumulation steps	5
Early-stop patience	5		

Supervised Finetuning: In the supervised finetuning, we adapt the pretrained encoder to perform downstream cell-type classification. We retain the same latent dimension ($D_z = 256$) and number of latents ($N = 128$) for architectural consistency, though all latent-related parameters remain frozen during finetuning. A single learning rate of 1×10^{-4} is applied to the trainable parameters (classification head, and optionally the final encoder block in partial finetuning).

Gradient clipping at 1.0 and weight decay of 5×10^{-4} are again used to stabilize optimization. Because finetuning runs under two regimes, partial finetuning and linear probing, we use batch sizes of 1 and 8, respectively, with 5 gradient accumulation steps to maintain memory efficiency. The finetuning setup includes 8 cross-attention heads (consistent with pretraining), although cross-attention blocks remain frozen. The classification head is a two-layer MLP with hidden dimension $h_l = 256$, which transforms the pooled encoder representation into cell-type logits. Similar to pretraining, we use early stopping with a patience of 5 epochs to avoid overfitting on smaller labeled datasets. Here, Table 6 presents the hyperparameters we have used for supervised finetuning.

Table 6: Hyperparameters for supervised finetuning (MAM).

Parameter	**Value**
Latent dimension D_z	256
# latents N	128
LR	1×10^{-4}
Gradient clipping	1.0
Batch size (partial, linear)	1, 8
Accumulation steps	5
Early-stop patience	5
Weight decay	5×10^{-4}
Cross-attn heads	8
Linear head hidden dim (h_l)	256

4.2.2 Computing Resources

To ensure fair and reproducible experimentation across all datasets and evaluation settings, we conducted all experiments on a shared high-performance compute environment. Each experiment was executed using 4 $\times$ NVIDIA A40 GPUs, each equipped with 48 GB VRAM. This setup was necessary to accommodate the large sequence lengths in scRNA-seq (up to 20K gene tokens), the memory demands of encoder–decoder foundation models, and the additional overhead introduced by the MAM cross-attention modules. All experiment groups including self-supervised pretraining, partial finetuning, and linear probing, were run using identical hardware to maintain consistency in runtime comparisons.

Table 7 summarizes the total wall-clock time required for each stage across all datasets. Because each configuration (masking scheme, injection strategy, finetuning mode) was repeated across 5 separate experiments, the reported runtimes represent the cumulative training and validation time for the full experimental sweep. The Hematopoietic Niche dataset required the longest runtime due to its larger size and higher sparsity, whereas PBMC68K and Heart datasets exhibited similar compute requirements.

Table 7: Compute resources and end-to-end runtime for every experiment group (MAM). Each experiment used 4 $\times$ NVIDIA A40 GPUs (48 GB each).

Dataset	Stage	# Experiments	Approx. Runtime
Hematopoietic niche	Pretrain	10	$\sim$4 weeks
	Finetune (partial)	10	$\sim$2 weeks
	Finetune (linear probe)	10	$\sim$1 week
	Total	**30**	**$\sim$7 weeks**
PBMC 68K	Pretrain	10	$\sim$3 weeks
	Finetune (partial)	10	$\sim$1 week
	Finetune (linear probe)	10	$\sim$1 week
	Total	**30**	**$\sim$5 weeks**
Heart	Pretrain	10	$\sim$3 weeks
	Finetune (partial)	10	$\sim$1 week
	Finetune (linear probe)	10	$\sim$1 week
	Total	**30**	**$\sim$5 weeks**

Overall, the compute costs highlight the challenge of training transformer-based FMs on full-length scRNA-seq inputs. Pretraining dominates total runtime due to the need to repeatedly reconstruct long gene sequences under dynamically generated MAM masks. In contrast, both finetuning and linear probing require substantially less time, as they operate on pretrained weights and involve significantly fewer trainable parameters. Despite these costs, the chosen hardware and parallelization strategy allowed us to execute all experiments within a feasible timeframe while maintaining controlled and replicable evaluation conditions.

4.2.3 Experiment Results

In this section, we evaluate the effectiveness of our Multinomial Attention Masking (MAM) framework across three single-cell RNA-seq datasets: Hematopoietic Niche, PBMC68K, and Heart. Our experiments

compare three major training regimes: Orig, which denotes finetuning or linear probing directly from the original pretrained backbone[1]; P, which refers to models that undergo an additional round of self-supervised pretraining on the target dataset; and FT or LP, which designate whether downstream evaluation is performed via partial finetuning of the model parameters or via linear probing with the encoder frozen.

For example, Orig+FT represents standard finetuning from the original checkpoint, whereas P+LP evaluates the impact of additional pretraining under a frozen-encoder classifier. We apply each setting to both random masking and MAM, enabling a controlled assessment of how masking strategy influences downstream accuracy. All reported numbers correspond to the mean and confidence interval across five independent runs, implemented via 5-fold cross-validation where one fold is held out for testing each time. This experimental design ensures a fair and statistically robust comparison across datasets, masking strategies, and training regimes.

MAM outperforms random masking: Across all three datasets (Hematopoietic Niche, PBMC68K, and Heart) MAM consistently provides higher downstream performance than uniform random masking. As shown in Table 8, under partial finetuning (P+FT), random masking reduces the F1-score relative to the original backbone, whereas MAM reliably recovers and surpasses these baselines. On the Hematopoietic Niche dataset, random masking (P+FT) reaches only 79.33% compared to 80.04% for Orig.+FT, while MAM with decoder injection achieves 81.33%, yielding a +2% absolute improvement.

A similar trend appears on PBMC68K, where random P+FT achieves 77.09% (Orig.+FT: 77.62%), but MAM with encoder injection reaches 79.58%, a +2–2.5% gain. Even on the less sparse Heart dataset, MAM still provides measurable improvements: the best P+FT configuration (decoder injection) achieves 92.57%, surpassing both Orig.+FT (91.63%) and random masking (91.90%).

On the Heart dataset, where baseline performance is already high, MAM still provides consistent improvements: random (P+FT) yields 91.90% (Orig.+FT: 91.63%), whereas MAM (decoder injection) achieves the highest score of 92.57%, showing that attention-guided masking remains beneficial even in cleaner, less sparse datasets. These results show that MAM not only mitigates the degradation caused by random masking during pretraining, but also boosts the model beyond the original backbone's performance.

For linear probing (LP), Table 9 further highlights the robustness of MAM. Random masking severely disrupts the representations learned during pretraining: on the Hematopoietic Niche dataset, P+LP with random masking collapses to 26.51%, far from the 77.36% achieved by Orig.+LP. In contrast, MAM (decoder injection) recovers the performance to 71.54%, restoring most of the backbone's accuracy despite the frozen encoder.

On PBMC68K, random masking drops to 65.55%, whereas MAM lifts performance to 68.43%. For the Heart dataset, although random masking degrades accuracy from 91.56% (Orig.+LP) to 89.99%, MAM configurations narrow this gap, with "Separate" injection reaching 90.13%. Overall, MAM consistently counteracts the disruptive effects of random masking, providing significantly more stable and biologically meaningful representations for downstream classification.

Table 8: Partial-finetune cell-type classification, comparing MAM (Ours) F1-score vs random masking. (FT: finetune, P: pretraining, Orig. = original backbone weights [1], all tokens: masked all attention-suggested tokens; 80/10/10: the standard random 80/10/10 scheme, Injection: which stream(s) received cross-attention output). Bolded numbers are the best. Results are reported as the mean ± confidence interval from 5-fold cross-validation on the test set, with each fold held out once.

Dataset	Masking	Injection	Train	F1-score
Hematopoietic	Random (80/10/10)	–	Orig.[1] + FT	80.04 ± 0.001
	Random (80/10/10)	–	P+FT	79.33 ± 0.001
	MAM (all tokens)	Both	P+FT	80.86 ± 0.001
	MAM (all tokens)	Separate	P+FT	81.31 ± 0.002
	MAM (all tokens)	Encoder	P+FT	81.10 ± 0.001
	MAM (all tokens)	Decoder	P+FT	**81.33 ± 0.002**
PBMC68K	Random (80/10/10)	–	Orig.[1] + FT	77.62 ± 0.004
	Random (80/10/10)	–	P+FT	77.09 ± 0.005
	MAM (all tokens)	Both	P+FT	79.02 ± 0.011
	MAM (all tokens)	Separate	P+FT	78.33 ± 0.007
	MAM (all tokens)	Encoder	P+FT	**79.58 ± 0.009**
	MAM (all tokens)	Decoder	P+FT	78.36 ± 0.007
Heart	Random (80/10/10)	–	Orig.[1] + FT	91.63 ± 0.004
	Random (80/10/10)	–	P+FT	90.90 ± 0.004
	MAM (all tokens)	Both	P+FT	92.37 ± 0.004
	MAM (all tokens)	Separate	P+FT	92.31 ± 0.005
	MAM (all tokens)	Encoder	P+FT	92.34 ± 0.005
	MAM (all tokens)	Decoder	P+FT	**92.57 ± 0.006**

Partial-finetuning (P+FT) vs. linear probing (P+LP): Across all three datasets, we observe that pretraining the backbone with *uniform random masking* (P+FT or P+LP) consistently pushes the model away from its original optimal initialization, resulting in degraded downstream performance.

For example, on the Hematopoietic Niche dataset (Table 8), the F1-score of the original backbone with full finetuning (Orig + FT) is 80.04%, but drops to 79.33% after pretraining with random masking (P+FT). The effect is even more severe under linear probing: Orig + LP achieves 77.36%, whereas P+LP collapses to only 26.51% (Table 9), showing that random masking perturbs the encoder representations in ways that the frozen linear head cannot recover from.

A similar trend appears on PBMC68K. With full finetuning, Orig + FT reaches 77.62%, while P+FT falls slightly to 77.09%. Under LP, the drop is again substantial: 70.67% (Orig + LP) decreases to 65.55% (P+LP). The Heart dataset follows the same pattern. Although the model is overall more robust on this dataset, pretraining with random masking still yields declines, from 91.63% (Orig + FT) to 90.90% (P+FT), and from 91.56% (Orig + LP) to 89.99% (P+LP). These results reinforce that random masking introduces an undesirable shift in the model's pretrained representations, undermining downstream classification performance.

In contrast, MAM consistently mitigates these issues. Under partial finetuning (P+FT), MAM not only avoids the degradation caused by random masking but often surpasses the original backbone. On the Hematopoietic Niche dataset, MAM reaches up to 81.33% (decoder injection), exceeding both P+FT (79.33%) and

Orig + FT (80.04%). On PBMC68K, MAM with encoder injection achieves 79.58%, a clear improvement over both baselines (77.09% and 77.62%). Even on the Heart dataset, where baseline scores are already high, MAM consistently improves performance, reaching 92.57% with decoder injection.

Under linear probing, MAM also recovers most of the performance lost by random masking. While P+LP disastrously collapses (e.g., 26.51% on Hematopoietic), MAM boosts performance to 71.54% (decoder injection), narrowing the gap to the original backbone (77.36%). On PBMC68K, MAM improves P+LP from 65.55% to 68.43%. Although gains on the Heart dataset are more modest (e.g., 90.13% vs. 89.99%), MAM consistently prevents the catastrophic degradation observed with random masking.

Overall, these findings indicate that while random masking destabilizes pretrained representations, especially harmful in linear probing, MAM preserves or enhances the backbone's internal structure. MAM (P+FT) consistently outperforms all baselines because both the backbone and MAM parameters can jointly adapt; in contrast, MAM (P+LP) is constrained by the frozen encoder, which explains the remaining gap to Orig + LP. Nevertheless, across all datasets and evaluation modes, MAM provides a significantly more stable and effective pretraining strategy than uniform random masking.

Table 9: Linear-probe cell-type classification (F1-score). Same notation as Table 8.

Dataset	Masking	Injection	Train	F1-score
Hematopoietic	Random (80/10/10)	–	Orig.[1] + LP	**77.36 ± 0.002**
	Random (80/10/10)	–	P+LP	26.51 ± 0.002
	MAM (all tokens)	Both	P+LP	70.55 ± 0.001
	MAM (all tokens)	Separate	P+LP	70.59 ± 0.001
	MAM (all tokens)	Encoder	P+LP	49.23 ± 0.002
	MAM (all tokens)	Decoder	P+LP	71.54 ± 0.001
PBMC68K	Random (80/10/10)	–	Orig.[1] + LP	**70.67 ± 0.005**
	Random (80/10/10)	–	P+LP	65.55 ± 0.010
	MAM (all tokens)	Both	P+LP	68.14 ± 0.010
	MAM (all tokens)	Separate	P+LP	68.25 ± 0.013
	MAM (all tokens)	Encoder	P+LP	68.17 ± 0.010
	MAM (all tokens)	Decoder	P+LP	68.43 ± 0.010
Heart	Random (80/10/10)	–	Orig.[1] + FT	**91.56 ± 0.004**
	Random (80/10/10)	–	P+FT	89.99 ± 0.006
	MAM (all tokens)	Both	P+FT	89.76 ± 0.007
	MAM (all tokens)	Separate	P+FT	90.13 ± 0.006
	MAM (all tokens)	Encoder	P+FT	89.61 ± 0.007
	MAM (all tokens)	Decoder	P+FT	89.57 ± 0.008

The impacts of context-injection schemes: Across all datasets, the choice of where to inject the latent context plays a meaningful role in the effectiveness of MAM. As shown in Table 8, under partial finetuning (P+FT), different datasets favor different injection routes.

For the Hematopoietic Niche dataset, injecting the latent summary into the decoder yields the strongest performance, reaching an F1-score of 81.33%, the highest among all configurations. For PBMC68K, however,

the encoder-only injection achieves the best result at 79.58%, suggesting that early-layer contextualization is more beneficial for this dataset. On the Heart dataset, the decoder again performs best, achieving 92.57%, slightly outperforming the other injection strategies.

The trend is similar, but more constrained, in the linear probing setting (Table 9), where only the classification head is trainable. In this frozen-encoder setting, the decoder, Both and Separate injections consistently recover the largest portion of the original backbone's performance across all datasets. For example, on the Hematopoietic dataset, MAM with decoder injection improves P+LP accuracy from 26.51% (random masking) to 71.54%. For the PBMC68K and Heart dataset, the gaps between injection schemes are smaller. On the PBMC68K, decoder injection similarly lifts the P+LP score from 65.55% to 68.43%. For the Heart dataset, "Separate" achieves the highest P+LP score (90.13%), followed closely by the other two multi-stream injections.

Overall, these results show that injecting latent-derived context meaningfully enhances MAM's effectiveness, but the optimal strategy is dataset-dependent. Decoder-side injection generally offers the most robust improvements, especially in P+FT, while combining injections (Both or Separate) provides competitive performance when the encoder is frozen, as in linear probing.

Masking style (80/10/10 vs. all tokens): Across all three datasets, the appendix tables (10–15) show a clear distinction between the behavior of random masking and MAM when switching from the standard 80/10/10 scheme to the "all tokens" variant. For random masking, replacing 80/10/10 with "all tokens" is often harmful.

On the Hematopoietic Niche dataset, for example, random (All tokens) under partial finetuning collapses to only 38.80% (Table 10), compared to 79.33% under the standard 80/10/10 masking. A similar degradation is observed in linear probing, where random (All tokens) reaches only 28.51% versus 26.51% for random (80/10/10), indicating no consistent gain. PBMC68K exhibits milder differences, but still shows no reliable improvement from switching to all-token random masking (e.g., 77.06% vs. 77.09% in P+FT; Table 11). In the Heart dataset, random (All tokens) performs roughly on par with 80/10/10 but provides no advantage (90.88% vs. 90.90%; Table 12).

Conversely, under the MAM framework, switching to "all tokens" consistently improves or maintains performance across all datasets and evaluation modes. For the Hematopoietic Niche dataset, MAM (All tokens) increases P+FT accuracy across every injection method, achieving 81.33% with Decoder injection (Table 10), the best result overall, while its MAM (80/10/10) counterpart lags far behind (e.g., 31.30–70.19%).

On PBMC68K, MAM (All tokens) achieves the best P+FT performance at 79.58% with Encoder injection (Table 11), outperforming all MAM (80/10/10) variants. The same trend holds for the Heart dataset, where MAM (All tokens) reaches 92.57% (Decoder injection), again exceeding all 80/10/10 configurations (Table 12). Linear probing results mirror these findings: while random all-token masking remains unstable, MAM (All tokens) consistently recovers most of the backbone accuracy, achieving 71.54% on Hematopoietic Niche, 68.43% on PBMC68K, and 90.13% on Heart (Tables 13–15).

Overall, the evidence indicates that although all-token masking can be fragile or even harmful under random masking, especially on sparse, heterogeneous datasets such as Hematopoietic Niche, it becomes highly beneficial when guided by attention-informed MAM scores. This highlights a core advantage of MAM: the ability to scale masking adaptively across the entire input space without destabilizing model training.

4.2.4 Biological Validation Results

While our quantitative experiments demonstrate that Multinomial Attention Masking (MAM) improves downstream classification performance, an important question remains: does MAM achieve this by learning biologically meaningful structure, or are its gains merely the result of a more complex masking procedure?

To address this, we perform a set of biological validation analyses aimed at understanding what genes MAM chooses to mask and why those choices matter biologically. Specifically, we examine (i) whether MAM preferentially masks highly expressed, information-rich genes, and (ii) whether the most frequently masked genes are enriched for known biological processes relevant to the underlying cell populations. Together, these analyses allow us to connect MAM's algorithmic behavior with interpretable biological mechanisms, showing that its improvements stem from learning to focus on genes that meaningfully define cellular identity and function.

Mask Frequency vs Gene Expression: A key question in understanding the behavior of our Multinomial Attention Masking (MAM) algorithm is whether it learns to selectively mask biologically meaningful genes rather than choosing positions arbitrarily. To investigate this, we analyzed masking behavior on a gene-by-gene basis. For every gene in the training set, we computed: (i) the total number of times the gene was selected for masking over all training iterations, and (ii) the gene's average expression value conditional on being masked.

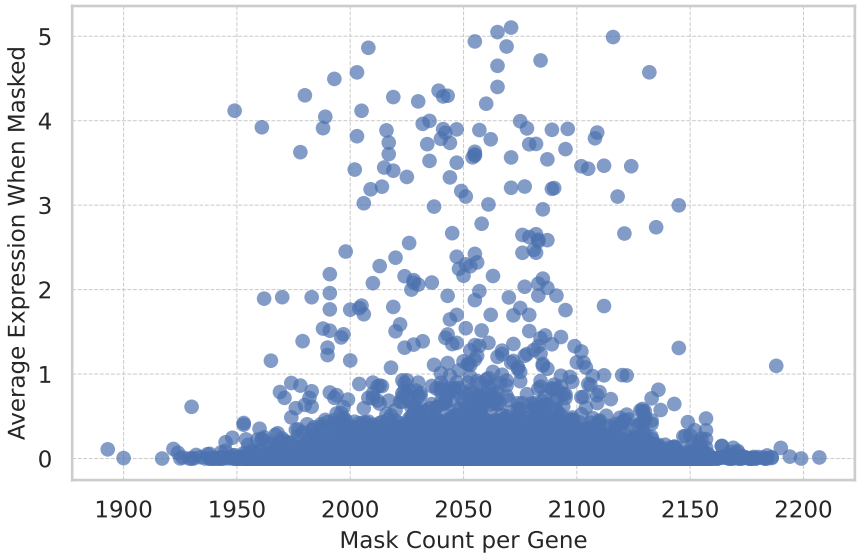

Figure 12: Scatter plot of each gene's masking frequency against its mean expression when masked (experiment 8, Appendix Table 11), illustrating that MAM preferentially targets genes with higher expression levels.

If masking were purely random, all genes would be masked at roughly similar frequencies over time, irrespective of their expression levels. In contrast, highly expressed genes typically carry stronger biological signal (reflecting cell identity, function, or activation state) so a learned masking strategy should prioritize masking these informative features to create more challenging reconstruction targets.

As shown in Fig. 12, we observe a clear and consistent positive correlation between masking frequency and mean expression level: genes with higher non-zero expression are masked substantially more often than genes expressed at or near zero. This trend appears across experiments (illustrated here with Experiment 8 in Appendix Table 11). This behavior strongly suggests that MAM does not simply sample positions uniformly. Instead, it learns an attention-driven policy that concentrates masking on the most informative coordinates of the transcriptome. Silent or weakly expressed genes, which provide little discriminative value, are masked far less frequently. Thus, MAM naturally prioritizes biologically relevant genes, leading to more effective self-supervised training and ultimately improved downstream performance.

GO Enrichment Analysis: To understand why our attention-driven masking strategy improves downstream model performance, we examined the biological relevance of the genes that MAM chooses to mask most frequently. As an illustrative case, we conducted a Gene Ontology (GO) enrichment analysis [247] on the PBMC68K dataset, focusing specifically on the subset of genes that were masked most often during pretraining. The top enriched biological processes are shown in Fig. 13.

The enrichment results reveal that several highly informative pathways such as lipid transport (GO:0006869), positive regulation of cholesterol transport (GO:0032376), phospholipid transport (GO:0015914), and ceramide transport (GO:0035627) which are among the most frequently masked by MAM. These processes play central roles in immune cell identity and function, including membrane organization, antigen presenta-

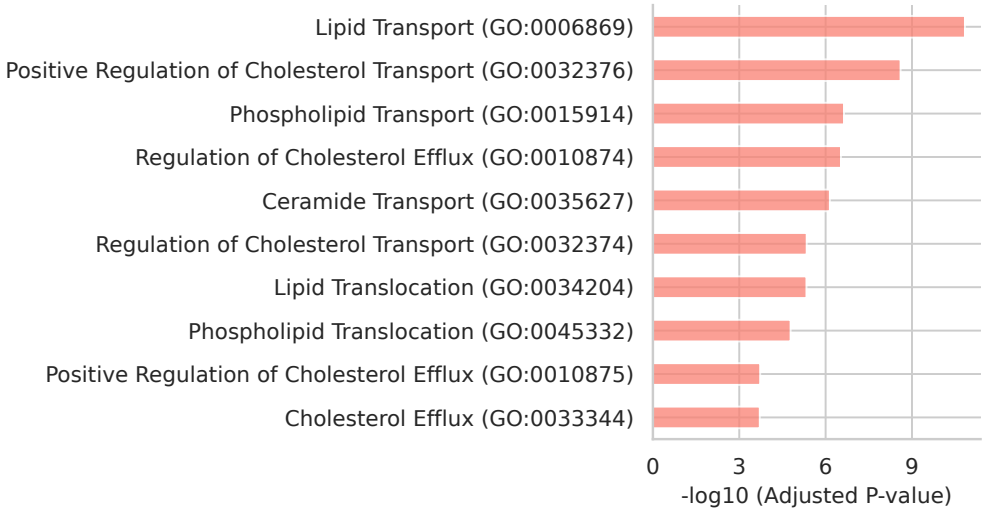

Figure 13: Top enriched GO biological processes among the most frequently masked genes in the PBMC68K dataset. The x-axis demonstrates the significance of each biological process (y-axis) derived from masked genes.

tion, vesicle trafficking, and cytokine signaling. Their prominence within the masked set strongly suggests that MAM selectively targets biologically meaningful genes rather than masking features indiscriminately.

In contrast, uniform random masking assigns equal probability to all genes regardless of their biological relevance or expression patterns, often forcing the model to reconstruct genes that are either uninformative or sparsely expressed. By masking highly informative genes instead, MAM encourages the model to learn deeper, more discriminative representations of cellular state: the decoder must reconstruct biologically important variation, and the encoder must capture structure that explains it. This targeted masking pressure ultimately leads to improved generalization and higher downstream performance.

Thus, the most frequently masked genes not only reflect underlying biological structure but also drive the model to internalize that structure more effectively. These findings highlight the value of integrating biologically guided masking strategies, such as MAM, into self-supervised learning pipelines for single-cell genomics, where sparsity and heterogeneity pose inherent challenges to uniform masking schemes.

4.2.5 Summary

In this objective, we introduced and evaluated the proposed Multinomial Attention Masking (MAM) algorithm as an alternative approach to the uniform random masking strategies commonly used in single-cell foundation models. Unlike random masking, which treats all genes equally and often ignores biologically informative positions, MAM leverages cross-attention scores from a set of learnable latent vectors to construct data-driven masking distributions that adapt to the underlying transcriptomic structure.

Our results demonstrate that MAM consistently outperforms random masking across all datasets and evaluation regimes, both by enhancing downstream F1-scores and by mitigating the representational degradation

introduced by random pretraining, especially severe under linear probing. We further show that the effectiveness of MAM depends on how latent-derived context is injected into the model, with decoder-side injection yielding the strongest and most stable improvements.

Importantly, biological validation analyses reveal that MAM preferentially masks highly expressed, information-rich genes and that these frequently masked genes are enriched for immune and lineage-relevant GO biological processes, confirming that MAM learns biologically meaningful masking patterns. Taken together, MAM demonstrates that attention-guided masking is a powerful and biologically principled enhancement to masked-autoencoder pretraining for scRNA-seq, improving robustness, interpretability, and downstream performance over existing random masking paradigms.

5 Conclusion

5.1 Summary of Findings

Across both objectives of our study, we conducted a comprehensive investigation into the performance, limitations, and improvement potential of single-cell foundation models. Objective 1 established a controlled benchmark for four widely used models revealing that although these models achieve strong aggregate performance, their generalization is highly dataset-dependent and sensitive to both biological context and data imbalance.

In particular, all models exhibited substantial drops in accuracy on clinically challenging AML relapse samples and on rare or transcriptionally subtle cell types such as CLP and Pre.B cells. These observations highlight persistent weaknesses inherited from the pretraining stage.

Objective 2 introduced the Multinomial Attention Masking (MAM) algorithm as a biologically informed alternative to random masking. By using latent cross-attention to learn gene importance dynamically, and by injecting this learned context back into the encoder–decoder architecture, MAM consistently improved masked-autoencoder pretraining across all three datasets studied.

Under both partial finetuning and linear probing, MAM not only mitigated the representational degradation caused by random masking but also surpassed the original pretrained backbone in many settings. Decoder-side context injection provided the strongest performance gains, though optimal injection pathways varied across datasets.

Biological validation analyses further demonstrated that MAM is not merely a more complex masking mechanism: it learns to prioritize highly expressed, information-rich genes and frequently masks genes enriched for immune and lineage-relevant GO processes. These findings provide strong evidence that MAM leverages biologically meaningful structure in the data to drive more effective self-supervised learning.

Taken together, our findings reveal two complementary insights. First, Objective 1 shows that although current single-cell foundation models provide strong baseline representations, their performance varies substantially across datasets, disease states, and rare cellular populations, underscoring persistent limitations rooted in existing pretraining practices.

Second, Objective 2 demonstrates that Multinomial Attention Masking (MAM) offers a biologically informed alternative to uniform random masking, improving model robustness, interpretability, and downstream accuracy through attention-guided gene prioritization and flexible context-injection mechanisms. While these objectives address different questions, they collectively illustrate both the challenges inherent to current single-cell FMs and the potential of targeted methodological innovations to enhance their pretraining dynamics.

5.2 Contributions and Impact

This work makes both methodological and empirical contributions to the development of single-cell foundation models. First, we provide one of the most systematic evaluations to date of four widely used single-cell FMs across three datasets spanning healthy, disease, and tissue-specific conditions. By fine-tuning each model under controlled settings, we show that their performance is far from uniform: they perform well on common cell types but struggle on rare lineages and in challenging clinical states such as AML relapse. These findings give a clearer picture of where current FMs succeed and where they continue to face limitations.

Building on this analysis, we introduce Multinomial Attention Masking (MAM), a new masking strategy designed specifically for the structure of scRNA-seq data. Instead of selecting masked genes at random, MAM uses attention over trainable latent vectors to identify which positions carry informative biological signal. The approach is lightweight, does not require a teacher model, and can be integrated easily into different FM architectures through several context-injection options.

Across all datasets and evaluation settings, MAM leads to more stable pretraining and higher downstream classification accuracy than uniform random masking. In many cases, it also surpasses the original pretrained backbone, showing that better masking alone can meaningfully strengthen model representations. Importantly, our biological validation demonstrates that MAM consistently prioritizes highly expressed and functionally relevant genes, indicating that it learns meaningful structure rather than relying on arbitrary or noisy patterns.

Overall, the contributions of this work provide both a clearer understanding of current FM behavior and a practical method for improving their pretraining dynamics. Our findings highlight the importance of biologically informed masking strategies and offer a path toward more robust and reliable single-cell foundation models.

5.3 Limitations

Although this work provides a broad evaluation of single-cell foundation models and introduces a biologically informed masking strategy, several limitations should be acknowledged. These limitations highlight areas where future studies could strengthen and extend our findings.

First, our downstream evaluation focused solely on cell-type classification, as it was the only task with reliable and consistent annotations across all datasets. While this benchmark is widely used and informative, it represents only one dimension of what foundation models are expected to handle. Other important tasks, such as trajectory inference, perturbation response prediction, regulatory network reconstruction, and multimodal integration, were beyond the scope of this study. A broader task portfolio would offer a more complete understanding of model capabilities and failure modes.

Second, although the three datasets used in our experiments span different tissues, they do not capture the

full breadth of clinical or demographic variability. Only the Hematopoietic Niche dataset includes detailed clinical timepoints, limiting our ability to analyze disease-specific performance in the other datasets.

Third, the effectiveness of MAM depends on the quality of the attention scores learned during pretraining. In settings where attention becomes diffuse or unstable, due to noise, sparsity, or batch effects, the resulting masking distribution may be less informative. Although MAM is lightweight and avoids the overhead of teacher–student frameworks, its attention computation still introduces additional cost compared to uniform masking, which may limit scalability for extremely large datasets.

Moreover, demographic information such as age, sex, race, and clinical background is largely unavailable or inconsistently reported across datasets. This limitation prevents a systematic analysis of demographic-specific performance and potential biases, and highlights the need for future studies to explore model behavior across more diverse and well-annotated populations.

Finally, our findings reflect a broader challenge in single-cell foundation models: they inherit biases from their pretraining dataset. As shown in the results, all evaluated models perform well on abundant and healthy cell types yet struggle with rare populations and clinically difficult states, such as AML relapse. While MAM improves pretraining robustness, it does not directly address data imbalance or biases. Progress in this area will require more diverse pretraining datasets and principled strategies for fairness-aware model development.

Overall, these limitations point toward several areas for future work, including expanding task diversity, integrating richer metadata, and developing methods to counteract biases in large-scale single-cell dataset.

5.4 Future Directions

Building on the insights obtained across both objectives of this study, several directions emerge for advancing single-cell foundation models and improving biologically grounded masking strategies. A natural next step is to explore a broader range of downstream tasks beyond cell-type classification. Tasks such as trajectory inference, perturbation response modeling, perturbation ranking, and gene regulatory network reconstruction would offer a more complete assessment of how pretraining and masking choices influence model generalization.

Another promising direction is the incorporation of clinical and demographic metadata during pretraining and evaluation. Datasets with richer annotations would make it possible to study fairness, subgroup performance, and domain shifts in greater depth. Evaluating models across disease progression, treatment response, and patient-level heterogeneity could reveal how well foundation models capture clinically meaningful variation, and whether masking strategies such as MAM can be adapted to these contexts.

Finally, addressing data imbalance and biases remains an open challenge. Future research may explore balanced pretraining strategies, synthetic minority cell augmentation, or curriculum-style masking schedules

that progressively emphasize challenging cell states. These approaches could reduce the systematic weaknesses observed in our study and move single-cell foundation models closer to equitable performance across rare, diseased, and clinically significant populations.

At the architectural level, applying MAM to a wider variety of pretrained models will help determine how masking interacts with different computational constraints and inductive biases. This may also motivate new variants of MAM tailored to compressed models or rank-based representations, where sparsity and gene ranking introduce unique challenges. Together, these directions outline a path toward the next generation of single-cell foundation models, models that are more robust, biologically grounded, and capable of generalizing across the full diversity of cellular and clinical settings.

Bibliography

[1] M. Hao, J. Gong, X. Zeng, C. Liu, Y. Guo, X. Cheng, T. Wang, J. Ma, X. Zhang, and L. Song, "Large-scale foundation model on single-cell transcriptomics," *Nature methods*, vol. 21, no. 8, pp. 1481–1491, 2024.

[2] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *arXiv preprint arXiv:1706.03762*, 2017.

[3] S. Ennis, A. Conforte, E. O'Reilly, J. S. Takanlu, T. Cichocka, S. P. Dhami, P. Nicholson, P. Krebs, P. Ó. Broin, and E. Szegezdi, "Cell-cell interactome of the hematopoietic niche and its changes in acute myeloid leukemia," *Iscience*, vol. 26, no. 6, 2023.

[4] A. A. Kolodziejczyk, J. K. Kim, V. Svensson, J. C. Marioni, and S. A. Teichmann, "The technology and biology of single-cell rna sequencing," *Molecular cell*, vol. 58, no. 4, pp. 610–620, 2015.

[5] A.-E. Saliba, A. J. Westermann, S. A. Gorski, and J. Vogel, "Single-cell rna-seq: advances and future challenges," *Nucleic Acids Research*, vol. 42, pp. 8845–8860, 07 2014.

[6] B. Van de Sande, J. S. Lee, E. Mutasa-Gottgens, B. Naughton, W. Bacon, J. Manning, Y. Wang, J. Pollard, M. Mendez, J. Hill, *et al.*, "Applications of single-cell rna sequencing in drug discovery and development," *Nature Reviews Drug Discovery*, vol. 22, no. 6, pp. 496–520, 2023.

[7] D. Jovic, X. Liang, H. Zeng, L. Lin, F. Xu, and Y. Luo, "Single-cell rna sequencing technologies and applications: A brief overview," *Clinical and translational medicine*, vol. 12, no. 3, p. e694, 2022.

[8] E. Hedlund and Q. Deng, "Single-cell rna sequencing: technical advancements and biological applications," *Molecular aspects of medicine*, vol. 59, pp. 36–46, 2018.

[9] A. T. Lun, D. J. McCarthy, and J. C. Marioni, "A step-by-step workflow for low-level analysis of single-cell rna-seq data with bioconductor," *F1000Research*, vol. 5, p. 2122, 2016.

[10] J. He, L. Lin, and J. Chen, "Practical bioinformatics pipelines for single-cell rna-seq data analysis," *Biophysics Reports*, vol. 8, no. 3, pp. 158–169, 2022.

[11] M. D. Luecken and F. J. Theis, "Current best practices in single-cell rna-seq analysis: a tutorial," *Molecular Systems Biology*, vol. 15, no. 6, p. e8746, 2019.

[12] Y. Hao, T. Stuart, M. H. Kowalski, *et al.*, "Dictionary learning for integrative, multimodal and scalable single-cell analysis," *Nature Biotechnology*, vol. 42, pp. 293–304, 2024.

[13] Y. Hao, S. Hao, E. Andersen-Nissen, W. M. Mauck, S. Zheng, A. Butler, M. J. Lee, A. J. Wilk, C. Darby, M. Zager, P. Hoffman, M. Stoeckius, E. Papalexi, E. P. Mimitou, J. Jain, A. Srivastava, T. Stuart, L. M. Fleming, B. Yeung, A. J. Rogers, J. M. McElrath, C. A. Blish, R. Gottardo, P. Smibert, and R. Satija, "Integrated analysis of multimodal single-cell data," *Cell*, vol. 184, no. 13, pp. 3573–3587.e29, 2021.

[14] T. Stuart, A. Butler, P. Hoffman, C. Hafemeister, E. Papalexi, I. Mauck, William M., Y. Hao, M. Stoeckius, P. Smibert, and R. Satija, "Comprehensive integration of single-cell data," *Cell*, vol. 177, pp. 1888–1902.e21, 2025/10/20 2019.

[15] A. Butler, P. Hoffman, P. Smibert, *et al.*, "Integrating single-cell transcriptomic data across different conditions, technologies, and species," *Nature Biotechnology*, vol. 36, pp. 411–420, 2018.

[16] R. Satija, J. Farrell, D. Gennert, *et al.*, "Spatial reconstruction of single-cell gene expression data," *Nature Biotechnology*, vol. 33, pp. 495–502, 2015.

[17] X. Xu, Y. Liang, M. Tang, J. Wang, X. Wang, Y. Li, and J. Wang, "Screni: Single-cell regulatory network inference through integrating scrna-seq and scatac-seq data," *Genomics, Proteomics & Bioinformatics*, p. qzaf060, 2025. Advance online publication.

[18] B. Yu, M. Sopic, and J. C. Sluimer, "Single-cell rna sequencing (scrna-seq) and its insights into cellular heterogeneity in atherosclerosis," *Vascular Pharmacology*, vol. 159, p. 107499, 2025.

[19] F. Ceccarelli, P. Liò, and S. B. Holden, "Annogcd: a generalized category discovery framework for automatic cell type annotation," *NAR Genomics and Bioinformatics*, vol. 6, no. 4, p. lqae166, 2024.

[20] C. Zheng, Y. Wang, Y. Cheng, X. Wang, H. Wei, I. King, and Y. Li, "scnovel: a scalable deep learning-based network for novel rare cell discovery in single-cell transcriptomics," *Briefings in Bioinformatics*, vol. 25, p. bbae112, 03 2024.

[21] S. Wang, H. Li, K. Zhang, H. Wu, S. Pang, W. Wu, L. Ye, J. Su, and Y. Zhang, "scsid: A lightweight algorithm for identifying rare cell types by capturing differential expression from single-cell sequencing data," *Computational and Structural Biotechnology Journal*, vol. 23, pp. 589–600, 2024.

[22] A. Gerniers, S. Nijssen, and P. Dupont, "sccross: efficient search for rare subpopulations across multiple single-cell samples," *Bioinformatics*, vol. 40, p. btae371, 06 2024.

[23] A. Gerniers, O. Bricard, and P. Dupont, "Microcellclust: mining rare and highly specific subpopulations from single-cell expression data," *Bioinformatics*, vol. 37, pp. 3220–3227, 04 2021.

[24] J. T. Leek, R. B. Scharpf, H. C. Bravo, D. Simcha, B. Langmead, W. E. Johnson, D. Geman, K. Baggerly, and R. A. Irizarry, "Tackling the widespread and critical impact of batch effects in high-throughput data," *Nature Reviews Genetics*, vol. 11, no. 10, pp. 733–739, 2010.

[25] J. Luo, M. Schumacher, A. Scherer, M. Sanhaji, C. Mayer, D. Brockmann, and H. A. Kestler, "A comparison of batch effect removal methods for enhancement of prediction performance using maqc-ii microarray gene expression data," *The Pharmacogenomics Journal*, vol. 10, no. 4, pp. 278–291, 2010.

[26] W. W. B. Goh, W. Wang, and L. Wong, "Why batch effects matter in omics data, and how to avoid them," *Trends in Biotechnology*, vol. 35, no. 6, pp. 498–507, 2017.

[27] V. Nygaard, E. A. Rodland, and E. Hovig, "Methods that remove batch effects while retaining group differences may lead to exaggerated confidence in downstream analyses," *Biostatistics*, vol. 17, no. 1, pp. 29–39, 2016.

[28] H. Al-Azzawi, B. Al-Shargabi, and A. A. Al-Marridi, "A comprehensive review of deep learning applications in single-cell rna sequencing analysis," *Briefings in Bioinformatics*, vol. 25, no. 2, p. bbae050, 2024.

[29] R. Lopez, J. Regier, M. B. Cole, M. I. Jordan, and N. Yosef, "Deep generative modeling for single-cell transcriptomics," *Nature Methods*, vol. 15, no. 12, pp. 1053–1058, 2018.

[30] C. V. Theodoris, L. Xiao, A. Chopra, M. D. Chaffin, Y. Perez, C. Miller, M. Lu, J. He, R. Melki, *et al.*, "Transfer learning with deep-learning-derived contextualized gene embeddings enables the interpretation of genetic variants," *Nature Genetics*, vol. 55, no. 4, pp. 625–636, 2023.

[31] H. Cui, C. Wang, H. Maan, K. Pang, F. Luo, N. Duan, and B. Wang, "scgpt: toward building a foundation model for single-cell multi-omics using generative ai," *Nature Methods*, vol. 21, no. 8, pp. 1470–1480, 2024.

[32] B. Hao, Z. Lu, Z. Gong, T. Ma, Z. Wei, H. Qu, Y. Li, J. Zhang, J.-B. Wang, and J. Li, "Geometric-enhanced and structure-aware self-supervised learning for single-cell analysis," *Nature Machine Intelligence*, vol. 5, no. 10, pp. 1183–1195, 2023.

[33] Wikipedia, "Transformer (deep learning architecture)," 2019. Accessed: 2025-10-20.

[34] Netguru, "Transformer models in natural language processing," *Netguru Blog*, 2025.

[35] IBM, "What is a transformer model?," *IBM Think*, 2025.

[36] Fabricated Knowledge, "Ai foundations part 1: Transformers, pre-training and fine-tuning," 2023.

[37] DesignGurus.io, "What is a transformer model architecture and why was it a breakthrough for nlp tasks," 2025.

[38] AWS, "What are foundation models? - generative ai," 2025.

[39] Wikipedia, "Generative pre-trained transformer," 2023.

[40] Neptune.ai, "10 things you need to know about bert and the transformer architecture," 2024.

[41] Zilliz, "What is bert (bidirectional encoder representations from transformers)," 2025.

[42] TechTarget, "What are masked language models (mlms)?," 2024.

[43] Indrasol, "Advancements in transformer architectures for large language model: from bert to gpt-3 and beyond," 2025.

[44] Hugging Face, "Next token prediction with gpt," 2025.

[45] J. Schneider, "Improving next tokens via second-to-last predictions with generate and refine," 2025.

[46] Reddit, "Why do we train language models with next word prediction?."

[47] G. Bachmann and V. Nagarajan, "The pitfalls of next-token prediction," 2025.

[48] Hugging Face, "Vitmae," 2022.

[49] V7 Labs, "Vision transformer: What it is & how it works [2024 guide]," 2025.

[50] W. Liu, F. Zhu, S. Ma, and C.-L. Liu, "Mspe: Multi-scale patch embedding prompts vision transformers to any resolution," in *Advances in Neural Information Processing Systems* (A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, eds.), vol. 37, pp. 29191–29212, Curran Associates, Inc., 2024.

[51] Viso.ai, "Vision transformer: A new era in image recognition," 2023.

[52] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16000–16009, 2022.

[53] Towards Data Science, "How to implement state-of-the-art masked autoencoders (mae)," 2025.

[54] Github/Lucidrains, "vit-pytorch: Implementation of vision transformer," 2020.

[55] N. Röhrich, A. Hoffmann, R. Nordsieck, E. Zarbali, and A. Javanmardi, "Masked autoencoder self pre-training for defect detection in microelectronics," 2025.

[56] Dive into Deep Learning, *Large-Scale Pretraining with Transformers*. 2012.

[57] Machine Learning Mastery, "A gentle introduction to positional encoding in transformer models, part 1," 2023.

[58] Towards Data Science, "Positional embeddings in transformers: A math guide to rope and alibi," 2025.

[59] GeeksforGeeks, "Positional encoding in transformers," 2024.

[60] Hugging Face, "You could have designed state of the art positional encoding," 2025.

[61] A. Kazemnejad, "Transformer architecture: The positional encoding," 2018.

[62] J. Alammar, "The illustrated transformer," 2018.

[63] R. Boiarsky, N. Singh, A. Buendia, G. Getz, and D. Sontag, "A deep dive into single-cell rna sequencing foundation models," *bioRxiv*, 2023.

[64] Towards Data Science, "Transformers explained visually (part 3): Multi-head attention deep dive," 2021.

[65] Polo Club, "Llm transformer model visually explained," 2016.

[66] Dive into Deep Learning, *Attention Mechanisms and Transformers*. 2017.

[67] S. Raschka, "Understanding and coding the self-attention mechanism," 2023.

[68] IBM, "What is an attention mechanism?," 2024.

[69] NVIDIA, "What is a transformer model?," 2024.

[70] NVIDIA, "How scaling laws drive smarter, more powerful ai," 2025.

[71] Learnius, "Encoder-decoder attention,"

[72] Stanford CS224N, "Fine-tuning bert for downstream tasks using multi-task learning," 2023.

[73] Wikipedia, "Bert (language model)," 2019.

[74] ApplyData.io, "Pre-training to fine-tuning of llms," 2025.

[75] Baeldung, "What are downstream tasks?," 2025.

[76] Wikipedia, "Fine-tuning (deep learning)," 2023.

[77] Encord, "Guide to transfer learning," 2024.

[78] A. Alajrami and N. Aletras, "How does the pre-training objective affect what large language models learn about linguistic properties?," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)* (S. Muresan, P. Nakov, and A. Villavicencio, eds.), (Dublin, Ireland), pp. 131–147, Association for Computational Linguistics, May 2022.

[79] IBM, "What are masked language models?," 2024.

[80] W. Liang and Y. Liang, "Bpdec: Unveiling the potential of masked language modeling decoder in bert pretraining," 2024.

[81] Hugging Face, "Masked language modeling," 2017.

[82] Shreydan, "Transformers pre-training: Masked language modeling," 2023.

[83] B. Shlegeris, F. Roger, L. Chan, and E. McLean, "Language models are better than humans at next-token prediction," *Transactions on Machine Learning Research*, 2024.

[84] Wonder's Lab, "Some thoughts on autoregressive models," 2025.

[85] IBM, "What is fine-tuning?," 2024.

[86] J. A. Raj, L. Qian, and Z. Ibrahim, "Fine-tuning – a transfer learning approach," 2024.

[87] Ultralytics, "Transfer learning: Definition, benefits & applications," 2025.

[88] L. Nowak, "Transfer learning and fine-tuning in machine learning," 2024.

[89] X. Jiang, X. Cheng, and Z. Li, "Why pre-training is beneficial for downstream classification tasks?," 2024.

[90] IBM, "What is few-shot learning?," 2024.

[91] NeurIPS, "Workshop on robustness of zero/few-shot learning in foundation models," 2023.

[92] Google Research, "Time series foundation models can be few-shot learners," 2025.

[93] DeepFA, "Zero-shot and few-shot learning: Learning with limited data," 2025.

[94] Microsoft, "Zero-shot and few-shot learning," 2025.

[95] Z. Xu, Z. Shi, J. Wei, F. Mu, Y. Li, and Y. Liang, "Towards few-shot adaptation of foundation models via multitask finetuning," in *The Twelfth International Conference on Learning Representations*, 2024.

[96] J. Kalfon, J. Samaran, G. Peyré, *et al.*, "scprint: pre-training on 50 million cells allows robust gene network predictions," *Nature Communications*, vol. 16, p. 3607, 2025.

[97] W. Han, Y. Cheng, J. Chen, H. Zhong, Z. Hu, S. Chen, L. Zong, L. Hong, T.-F. Chan, I. King, X. Gao, and Y. Li, "Self-supervised contrastive learning for integrative single cell rna-seq data analysis," *Briefings in Bioinformatics*, vol. 23, p. bbac377, 09 2022.

[98] Y. Zeng, J. Xie, N. Shangguan, *et al.*, "Cellfm: a large-scale foundation model pre-trained on transcriptomics of 100 million human cells," *Nature Communications*, vol. 16, p. 4679, 2025.

[99] Y. Zhang *et al.*, "Efficient and effective foundation model on single cell data," *arXiv preprint arXiv:2504.16956*, 2025.

[100] S. Baek, K. Song, and I. Lee, "Single-cell foundation models: bringing artificial intelligence into cell biology," *Experimental & Molecular Medicine*, vol. 57, pp. 2169–2181, 2025.

[101] Y. Zeng and Y. Yang, "Foundation model: A new era for plant single-cell genomics," *Genomics, Proteomics & Bioinformatics*, vol. 23, no. 3, p. qzaf059, 2025.

[102] T. Liu, K. Li, Y. Wang, H. Li, and H. Zhao, "Evaluating the utilities of foundation models in single-cell data analysis," *bioRxiv*, 2024. Preprint.

[103] Y. Ji, Z. Zhou, H. Liu, and R. V. Davuluri, "Dnabert: pre-trained bidirectional encoder representations from transformers for dna-language in genome," *Bioinformatics*, vol. 37, no. 15, pp. 2112–2120, 2021.

[104] S. Palayew, B. WANG, and G. D. Bader, "scMPT: towards applying large language models to complement single-cell foundation models," 2024.

[105] F. Yang, W. Wang, F. Wang, Y. Fang, D. Tang, J. Huang, H. Lu, and J. Yao, "scbert as a large-scale pretrained deep language model for cell type annotation of single-cell rna-seq data," *Nature Machine Intelligence*, vol. 4, no. 10, pp. 852–866, 2022.

[106] ClinicalML, "sc-foundation-eval: Code for evaluating single-cell foundation models," 2023.

[107] H. Dalla-Torre, L. Gonzalez, J. Mendoza-Revilla, N. L. Carranza, A. H. Grzywaczewski, F. Oteri, C. Dallago, E. Trop, B. P. de Almeida, H. Sirelkhatim, *et al.*, "Nucleotide transformer: building and evaluating robust foundation models for human genomics," *Nature Methods*, 2024.

[108] P. Ramprasad, N. Pai, and W. Pan, "Enhancing personalized gene expression prediction from dna sequences using genomic foundation models," *HGG Advances*, vol. 5, no. 4, p. 100347, 2024.

[109] G. Dong, Y. Wu, L. Huang, F. Li, and F. Zhou, "Texcnn: Leveraging pre-trained models to predict gene expression from genomic sequences," *Genes*, vol. 15, no. 12, 2024.

[110] N. Ghosh, D. Santoni, I. Saha, and G. Felici, "A review on the applications of transformer-based language models for nucleotide sequence analysis," *Computational and Structural Biotechnology Journal*, vol. 27, p. 1244–1254, 2025.

[111] InstaDeep, "nucleotide-transformer: Foundation models for genomics," 2023.

[112] vZ. Avsec, V. Agarwal, D. Visentin, J. R. Ledsam, A. Grabska-Barwinska, K. R. Taylor, Y. Assael, J. Jumper, P. Kohli, and D. R. Kelley, "Effective gene expression prediction from sequence by integrating long-range interactions," *Nature Methods*, vol. 18, no. 10, pp. 1196–1203, 2021.

[113] G. Cao, H. Chao, W. Zheng, Y. Lan, K. Lu, Y. Wang, M. Chen, H. Zhang, and D. Chen, "scplantllm: A foundation model for exploring single-cell expression atlases in plants," *Genomics, Proteomics Bioinformatics*, vol. 23, p. qzaf024, 03 2025.

[114] Valence Labs, "A primer on scrna-seq foundation models," 2024.

[115] G. Csendes, G. Sanz, K. Z. Szalay, and B. Szalai, "Benchmarking foundation cell models for post-perturbation rna-seq prediction," *BMC Genomics*, vol. 26, no. 1, p. 393, 2025.

[116] R. Jiang, T. Sun, D. Song, and J. J. Li, "Zeros in scrna-seq a universal analytical challenge," *bioRxiv*, 2020.

[117] Y. Imoto, T. Nakamura, E. G. Escolar, M. Yoshiwaki, Y. Kojima, Y. Yabuta, Y. Katou, T. Yamamoto, Y. Hiraoka, and M. Saitou, "Resolution of the curse of dimensionality in single-cell rna sequencing data analysis," *Life Science Alliance*, vol. 5, no. 12, 2022.

[118] A. Pavel *et al.*, "The impact of dropouts in scrnaseq dense neighborhood clustering," *Computational and Structural Biotechnology Journal*, vol. 23, pp. 1140–1151, 2025.

[119] D. Ran *et al.*, "scdoc: correcting drop-out events in single-cell rna-seq data," *Bioinformatics*, vol. 36, no. 15, pp. 4233–4239, 2020.

[120] P. Qiu, "Embracing the dropouts in single-cell rna-seq analysis," *Nature Communications*, vol. 11, p. 1169, 2020.

[121] N. Khetan *et al.*, "Single-cell rna sequencing algorithms underestimate noise," *Cell Genomics*, vol. 4, 2024.

[122] M. Shi *et al.*, "Addressing scalability and managing sparsity and dropout in scrna-seq clustering," *BMC Bioinformatics*, vol. 26, p. 8, 2025.

[123] T. Li *et al.*, "An overview of computational methods in single-cell rna sequencing data analysis," *Briefings in Bioinformatics*, vol. 26, no. 3, p. bbaf207, 2025.

[124] V. L. Zogopoulos *et al.*, "Single-cell rna sequencing data dimensionality reduction and visualization," *World Academy of Sciences Journal*, vol. 7, p. 315, 2025.

[125] Y. Zhang, Y. Wang, X. Liu, and X. Feng, "Pbimpute: Precise zero discrimination and balanced imputation in single-cell rna sequencing data," *Journal of Chemical Information and Modeling*, vol. 65, no. 5, pp. 2670–2684, 2025. PMID: 39957720.

[126] Y. Imoto *et al.*, "Resolution of the curse of dimensionality in single-cell rna sequencing data analysis," *PubMed*, 2022. PMID: 35944930.

[127] L. Jiang *et al.*, "Accounting for technical noise in bayesian graphical models for single-cell rna sequencing data," *Biometrics*, 2022.

[128] N. Erfanian, A. A. Heydari, A. M. Feriz, P. Iañez, A. Derakhshani, M. Ghasemigol, M. Farahpour, S. M. Razavi, S. Nasseri, H. Safarpour, *et al.*, "Deep learning applications in single-cell genomics and transcriptomics data analysis," *Biomedicine & Pharmacotherapy*, vol. 165, p. 115077, 2023.

[129] S. Wang *et al.*, "Federated learning for cell type classification with scrna-seq data," *Briefings in Bioinformatics*, 2024.

[130] T. Li *et al.*, "An overview of computational methods in single-cell rna sequencing," *Briefings in Bioinformatics*, 2025.

[131] H. Wang *et al.*, "Learning discriminative and structural samples for rare cell types identification in scrna-seq data," *PubMed*, 2022. PMID: 35914950.

[132] Y. Cheng *et al.*, "A scalable sparse neural network framework for rare cell type annotation," *Communications Biology*, vol. 6, p. 552, 2023.

[133] H. Kim *et al.*, "sclens: data-driven signal detection for unbiased scrna-seq data analysis," *Nature Communications*, vol. 15, p. 3520, 2024.

[134] J. D. Silverman *et al.*, "Naught all zeros in sequence count data are the same," *Computational and Structural Biotechnology Journal*, vol. 18, pp. 2789–2798, 2020.

[135] Z. Fang, R. Zheng, and M. Li, "scmae: a masked autoencoder for single-cell rna-seq clustering," *Bioinformatics*, vol. 40, no. 1, p. btae020, 2024.

[136] J. Yang *et al.*, "Crafted experiments to evaluate feature selection methods for scrna-seq clustering," *NAR Genomics and Bioinformatics*, vol. 7, no. 1, p. lqaf023, 2025.

[137] Y. Zhang *et al.*, "Sieve: identifying robust single cell variable genes for scrna-seq," *BMC Genomics*, vol. 22, p. 332, 2021.

[138] L. Zappia *et al.*, "Feature selection methods affect the performance of integrating scrna-seq samples," *Nature Methods*, 2025.

[139] S. Kommu *et al.*, "Prediction of gene regulatory connections with joint single-cell foundation models and graph-based learning," *Bioinformatics*, vol. 41, 2025.

[140] OSCA, "Feature selection — basics of single-cell analysis," 2021.

[141] Y. Imoto *et al.*, "Accurate highly variable gene selection using recode in scrna-seq data analysis," *bioRxiv*, 2025.

[142] Single-cell Best Practices, "Feature selection," 2019.

[143] N. M. Xi and J. J. Li, "Exploring the optimization of autoencoder design for imputing scrna-seq data," *Computational and Structural Biotechnology Journal*, vol. 21, pp. 4079–4095, 2023.

[144] Z. Hong *et al.*, "The diminishing returns of masked language models to science," pp. 1270–1283, 2023.

[145] J. K. Arora *et al.*, "Computational workflow for investigating highly variable genes in single-cell rna-seq data," *STAR Protocols*, vol. 4, p. 102439, 2023.

[146] G. Molla Desta *et al.*, "Advancements in single-cell rna sequencing and spatial transcriptomics," *Frontiers in Biophysics*, 2025.

[147] N. Karenna *et al.*, "Focused learning by antibody language models using biologically-informed masking strategies," *Nature Communications*, vol. 15, p. 9400, 2024.

[148] J. C. Timoneda *et al.*, "Random and selective masking in transformer models: Representation learning from a sparsely annotated image dataset," *Patterns*, vol. 6, p. 101093, 2025.

[149] J. Xu *et al.*, "Stgrns: an interpretable transformer-based method for inferring gene regulatory networks from single-cell transcriptomic data," *Bioinformatics*, vol. 39, no. 4, p. btad165, 2023.

[150] Q. Li *et al.*, "Progress and opportunities of foundation models in bioinformatics," *Briefings in Bioinformatics*, vol. 25, no. 6, p. bbae548, 2024.

[151] A. Lin *et al.*, "A comprehensive review of large language models in bioinformatics," *Briefings in Bioinformatics*, vol. 26, no. 4, p. bbaf357, 2025.

[152] P. Qiu *et al.*, "Biollm: A standardized framework for integrating large language models in bioinformatics," *Patterns*, vol. 6, p. 101194, 2025.

[153] J. Smith *et al.*, "Interpreting attention mechanisms in genomic transformer models," *bioRxiv*, 2025.

[154] O. A. Sarumi *et al.*, "Large language models and their applications in bioinformatics: A systematic review," *Computational and Structural Biotechnology Journal*, vol. 23, pp. 2983–3003, 2024.

[155] J. Chen *et al.*, "Transformer for one stop interpretable cell type annotation," *Nature Communications*, vol. 14, p. 223, 2023.

[156] Y. Zhang *et al.*, "scmaskgan: masked multi-scale cnn and attention mechanism for single-cell imputation," *BMC Bioinformatics*, vol. 26, p. 187, 2025.

[157] L. Lin *et al.*, "Multi-scale scrna-seq sub-vector completion transformer," pp. 5954–5962, 2024.

[158] Y. Bitton *et al.*, "Data efficient masked language modeling for vision and language," pp. 3043–3059, 2021.

[159] J. Xie, Y. Lee, A. S. Chen, and C. Finn, "Self-guided masked autoencoders for domain-agnostic self-supervised learning," *arXiv preprint arXiv:2402.14789*, 2024.

[160] Z. Cui, T. Xu, J. Wang, Y. Liao, and Y. Wang, "Geneformer: Learned gene compression using transformer-based context modeling," pp. 8035–8039, 2024.

[161] D. J. McCarthy, K. R. Campbell, A. T. Lun, and Q. F. Wills, "Scater: pre-processing, quality control, normalization and visualization of single-cell rna-seq data in r," *Bioinformatics*, vol. 33, no. 8, pp. 1179–1186, 2017.

[162] T. Ilicic, J. K. Kim, A. A. Kolodziejczyk, F. O. Bagger, D. J. McCarthy, J. C. Marioni, and S. A. Teichmann, "Classification of low quality cells from single-cell rna-seq data," *Genome Biology*, vol. 17, p. 29, 2016.

[163] M. B. Cole, D. Risso, A. Wagner, D. DeTomaso, J. Ngai, E. Purdom, S. Dudoit, and N. Yosef, "Performance assessment and selection of normalization procedures for single-cell rna-seq," *Cell Systems*, vol. 8, no. 4, pp. 315–328, 2019.

[164] C. A. Vallejos, D. Risso, A. Scialdone, S. Dudoit, and J. C. Marioni, "Normalizing single-cell rna sequencing data: challenges and opportunities," *Nature Methods*, vol. 14, no. 6, pp. 565–571, 2017.

[165] A. T. Lun, K. Bach, and J. C. Marioni, "Pooling across cells to normalize single-cell rna sequencing data with many zero counts," *Genome Biology*, vol. 17, p. 75, 2016.

[166] C. Hafemeister and R. Satija, "Normalization and variance stabilization of single-cell rna-seq data using regularized negative binomial regression," *Genome Biology*, vol. 20, p. 296, 2019.

[167] P. Brennecke, S. Anders, J. K. Kim, A. A. KorAodziejczyk, X. Zhang, V. Proserpio, B. Baying, V. Benes, S. A. Teichmann, J. C. Marioni, and M. G. Heisler, "Accounting for technical noise in single-cell rna-seq experiments," *Nature Methods*, vol. 10, no. 11, pp. 1093–1095, 2013.

[168] T. S. Andrews, V. Y. Kiselev, D. McCarthy, and M. Hemberg, "Tutorial: guidelines for the computational analysis of single-cell rna sequencing data," *Nature Protocols*, vol. 16, pp. 1–9, 2021.

[169] T. S. Andrews and M. Hemberg, "M3drop: dropout-based feature selection for scrnaseq," *Bioinformatics*, vol. 35, no. 16, pp. 2865–2867, 2019.

[170] B. Liu, C. Li, Z. Li, D. Wang, X. Ren, and Z. Zhang, "An entropy-based metric for assessing the purity of single cell populations," *Nature Communications*, vol. 11, p. 3155, 2020.

[171] J. Cho, H. Hwang, D. Baek, J. Bae, H. Lee, J. Yang, J. Choi, and I. Jung, "Characterizing efficient feature selection for single-cell rna sequencing," *Genome Biology*, vol. 25, p. 169, 2024.

[172] L. Zappia, A. Lun, R. Amezquita, *et al.*, "Feature selection methods affect the performance of reference-based single-cell rna-seq analysis," *Nature Methods*, vol. 22, pp. 463–473, 2025.

[173] S. Sun, J. Zhu, Y. Ma, and X. Zhou, "Accuracy, robustness and scalability of dimensionality reduction methods for single-cell rna-seq analysis," *Genome Biology*, vol. 20, p. 269, 2019.

[174] R. Xiang, W. Wang, L. Yang, S. Wang, C. Xu, and X. Chen, "A comparison for dimensionality reduction methods of single-cell rna-seq data," *Frontiers in Genetics*, vol. 12, p. 646936, 2021.

[175] L. van der Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, 2008.

[176] L. McInnes, J. Healy, and J. Melville, "Umap: Uniform manifold approximation and projection for dimension reduction," *arXiv preprint arXiv:1802.03426*, 2018.

[177] E. Becht, L. McInnes, J. Healy, C.-A. Dutertre, I. W. Kwok, L. G. Ng, F. Ginhoux, and E. W. Newell, "Dimensionality reduction for visualizing single-cell data using umap," *Nature Biotechnology*, vol. 37, pp. 38–44, 2019.

[178] J. Ding, A. Condon, and S. P. Shah, "Interpretable dimensionality reduction of single cell transcriptome data with deep generative models," *Nature Communications*, vol. 9, p. 2002, 2018.

[179] V. Y. Kiselev, T. S. Andrews, and M. Hemberg, "Challenges in unsupervised clustering of single-cell rna-seq data," *Nature Reviews Genetics*, vol. 20, pp. 273–282, 2019.

[180] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2008, p. P10008, 2008.

[181] V. A. Traag, L. Waltman, and N. J. Van Eck, "From louvain to leiden: guaranteeing well-connected communities," *Scientific Reports*, vol. 9, p. 5233, 2019.

[182] V. Y. Kiselev, K. Kirschner, M. T. Schaub, T. Andrews, A. Yiu, T. Chandra, K. N. Natarajan, W. Reik, M. Barahona, A. R. Green, and M. Hemberg, "Sc3: consensus clustering of single-cell rna-seq data," *Nature Methods*, vol. 14, pp. 483–486, 2017.

[183] V. Nasrollahi, R. Alhajj, and J. G. Rokne, "Cell clustering for spatial transcriptomics data with graph neural networks," *Nature Machine Intelligence*, vol. 6, pp. 137–150, 2024.

[184] G. Pasquini, J. E. Rojo Arias, P. SchÃ¤fer, and V. Busskamp, "Automated methods for cell type annotation on scrna-seq data," *Computational and Structural Biotechnology Journal*, vol. 19, pp. 961–969, 2021.

[185] M. Cheng, C. Xiao, Y. Wang, J. He, Q. Liu, G. Zhao, B. Zhang, Q. Yang, G. Liu, L. Qian, Y. Wang, and J. He, "A review of computational methods for clustering genes with similar biological functions," *Computers in Biology and Medicine*, vol. 153, p. 106469, 2023.

[186] D. Aran, A. P. Looney, L. Liu, E. Wu, V. Fong, A. Hsu, S. Chak, R. P. Naikawadi, P. J. Wolters, A. R. Abate, A. J. Butte, and M. Bhattacharya, "Reference-based analysis of lung single-cell sequencing reveals a transitional profibrotic macrophage," *Nature Immunology*, vol. 20, pp. 163–172, 2019.

[187] A. W. Zhang, C. O'Flanagan, E. A. Chavez, J. L. Lim, N. Ceglia, A. McPherson, M. Wiens, P. Walters, T. Chan, B. Hewitson, *et al.*, "Probabilistic cell-type assignment of single-cell rna-seq for tumor microenvironment profiling," *Nature Methods*, vol. 16, pp. 1007–1015, 2019.

[188] J. Alquicira-Hernandez, A. Sathe, H. P. Ji, Q. Nguyen, and J. E. Powell, "scpred: accurate supervised method for cell-type classification from single-cell rna-seq data," *Genome Biology*, vol. 20, p. 264, 2019.

[189] Y. Tan and P. Cahan, "Singlecellnet: A computational tool to classify single cell rna-seq data across platforms and across species," *Cell Systems*, vol. 9, no. 2, pp. 207–213, 2019.

[190] F. Ma and M. Pellegrini, "Actinn: automated identification of cell types in single cell rna sequencing," *Bioinformatics*, vol. 36, no. 2, pp. 533–538, 2020.

[191] M. Lotfollahi, M. Naghipourfar, M. D. Luecken, M. Khajavi, M. BÃ¼ttner, M. Wagenstetter, Avsec, A. Gayoso, N. Yosef, M. Interlandi, *et al.*, "Mapping single-cell data to reference atlases by transfer learning," *Nature Biotechnology*, vol. 40, pp. 121–130, 2022.

[192] G. Eraslan, Avsec, J. Gagneur, and F. J. Theis, "Deep learning: new computational modelling techniques for genomics," *Nature Reviews Genetics*, vol. 20, pp. 389–403, 2019.

[193] M. Brendel, Q. Su, C. Baccin, F. Oswald, S. SchÃ¤dler, F. Denk, A.-C. Hauschild, and S. Baumeister, "Application of deep learning on single-cell rna sequencing data analysis: A review," *Genomics, Proteomics*
Bioinformatics, vol. 20, no. 5, pp. 814–835, 2022.

[194] D. Talwar, A. Mongia, D. Sengupta, and A. Majumdar, "Autoimpute: Autoencoder based imputation of single-cell rna-seq data," *Scientific Reports*, vol. 8, p. 16329, 2018.

[195] M. B. Badsha, B. Li, Y.-H. Liu, Y.-Y. Ni, M. Collier, A. Kipnis, and A. Q. Fu, "Imputation of single-cell gene expression with an autoencoder neural network," *Quantitative Biology*, vol. 8, pp. 78–94, 2020.

[196] Y. Xi, B. Huang, Y. Su, W. Li, J. Yang, J. Chen, L. Chen, N. Li, Q. Jiang, Z. Huang, T. Lei, Q. Hou, G. Qin, W. Zhou, W. Li, and S. Zhang, "Exploring optimal autoencoder design for imputing single-cell rna sequencing data," *Biology Direct*, vol. 18, p. 12, 2023.

[197] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2014.

[198] A. Gayoso, R. Lopez, G. Xing, P. Boyeau, V. Valiollah Pour Amiri, J. Hong, K. Wu, M. Jayasuriya, E. Mehlman, M. Langevin, *et al.*, "A python library for probabilistic analysis of single-cell omics data," *Nature Biotechnology*, vol. 40, pp. 163–166, 2022.

[199] M. Carilli, S. Ghose, A. Bhagwat, R. Lopez, A. Regev, A. Mani, N. Yosef, *et al.*, "Biophysical modeling with variational autoencoders for bimodal, single-cell rna sequencing data," *Nature Methods*, vol. 20, pp. 1452–1462, 2023.

[200] V. Svensson, A. Gayoso, N. Yosef, and L. Pachter, "Interpretable factor models of single-cell rna-seq via variational autoencoders," *Bioinformatics*, vol. 36, no. 11, pp. 3418–3421, 2020.

[201] A. Gayoso, Z. Steier, R. Lopez, J. Regier, K. L. Nazor, A. Streets, and N. Yosef, "Joint probabilistic modeling of single-cell multi-omic data with totalvi," *Nature Methods*, vol. 18, pp. 272–282, 2021.

[202] A. Sarkar and M. Stephens, "Separating measurement and expression models clarifies confusion in single-cell rna sequencing analysis," *Nature Genetics*, vol. 53, pp. 770–777, 2021.

[203] J. Wang, A. Ma, Y. Chang, J. Gong, Y. Jiang, R. Qi, C. Wang, H. Fu, Q. Ma, and D. Xu, "scgnn is a novel graph neural network framework for single-cell rna-seq analyses," *Nature Communications*, vol. 12, p. 1882, 2021.

[204] H. Li, J. Janssens, M. De Waegeneer, S. S. Kolluru, K. Vandereyken, R. Seurinck, J. Verdonck, and Y. Saeys, "Fly cell atlas: A single-nucleus transcriptomic atlas of the adult fruit fly," *Science*, vol. 375, p. eabk2432, 2022.

[205] Y. Mao, Y. Zhang, Y. S. Wang, *et al.*, "Predicting gene regulatory links from single-cell rna-seq data using graph neural networks," *Briefings in Bioinformatics*, vol. 24, no. 1, p. bbac559, 2023.

[206] Q. Li, H. Xu, *et al.*, "scgraph: a graph neural network-based approach to automatically identify cell types," *Bioinformatics*, vol. 39, no. 1, p. btac842, 2023.

[207] W. Han, Y. Cheng, J. Chen, H. Zhong, Z. Hu, S. Chen, L. Zong, L. Hong, W. Xue, G. Li, H. Chen, S. Wang, Z. Qiao, K. Yang, C. Gao, Y. Guo, W. Li, H. Zheng, Z. Chen, K. Chen, Z. Yang, J. Peng, C. Chen, X. Li, K. Liu, D.-P. Lin, Z. Zhou, M. Fang, R. Chen, and X. Chen, "Self-supervised contrastive learning for integrative single cell rna-seq data analysis," *Briefings in Bioinformatics*, vol. 23, no. 5, p. bbac377, 2022.

[208] B. Du, Z. Niu, Y. Xie, Y. Tian, X. Yao, X. Wei, Q. Yang, and J. Chen, "Scccl: A contrastive clustering framework for single-cell rna-seq data analysis," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 20, no. 5, pp. 3029–3040, 2023.

[209] Y. Tan, Y. Xu, Z. Zhang, *et al.*, "scamac: self-supervised contrastive learning with adaptive multi-scale attention mechanism for clustering of single-cell rna-seq data," *Briefings in Bioinformatics*, vol. 25, no. 1, p. bbad497, 2024.

[210] Y. Heryanto *et al.*, "Predicting cell types with deep learning on scrna-seq datasets," *BMC Bioinformatics*, vol. 25, p. 89, 2024.

[211] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Kaiser, and I. Polosukhin, "Attention is all you need," vol. 30, 2017.

[212] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2019.

[213] R. Boiarsky, J. Buenrostro, and L. Pinello, "Deep learning foundations for single-cell genomics: A critical analysis," *Nature Methods*, vol. 20, pp. 1439–1447, 2023.

[214] C. Zhao, Y. Zeng, K. Sun, H. Liu, Y. Chen, and F. Zhou, "Cellplm: Pre-training of cell language model beyond single cells," *iScience*, vol. 27, p. 109563, 2024.

[215] H. Ye, H. Ye, H. Ji, and Z. S. Wang, "Cellfm: a large-scale foundation model pre-trained on 100 million human cells," *bioRxiv*, 2024.

[216] Y. Rosen, Y. Roohani, A. Agarwal, L. Samson, J. J. Kang, A. Kamath, J. Jimenez, A. Arora, K. Watanabe, S. B. Montgomery, S. R. Quake, P. A. Beachy, and J. Zou, "Universal cell embeddings: A foundation model for cell biology," *bioRxiv*, 2024.

[217] S. E. Antonsson, A. Henningsson, E. SkÃ¶ldberg, L. Fagerberg, R. Pinto, K. M. Persson, A. Roos, U. Gullberg, and G. JÃ¶nsson, "Batch correction methods used in single-cell rna sequencing analysis," *BMC Genomics*, vol. 26, p. 39, 2025.

[218] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, *et al.*, "Language models are few-shot learners," *Advances in Neural Information Processing Systems*, vol. 33, pp. 1877–1901, 2020.

[219] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," *OpenAI Blog*, 2018.

[220] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2021.

[221] J. Howard and S. Ruder, "Universal language model fine-tuning for text classification," *arXiv preprint arXiv:1801.06146*, 2018.

[222] N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. De Laroussilhe, A. Gesmundo, M. Attariyan, and S. Gelly, "Parameter-efficient transfer learning for nlp," pp. 2790–2799, 2019.

[223] B. Lester, R. Al-Rfou, and N. Constant, "The power of scale for parameter-efficient prompt tuning," *arXiv preprint arXiv:2104.08691*, 2021.

[224] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "Lora: Low-rank adaptation of large language models," *arXiv preprint arXiv:2106.09685*, 2022.

[225] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[226] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He, "A comprehensive survey on transfer learning," *Proceedings of the IEEE*, vol. 109, no. 1, pp. 43–76, 2021.

[227] B. Zoph, G. Ghiasi, T.-Y. Lin, Y. Cui, H. Liu, E. D. Cubuk, and Q. V. Le, "Rethinking pre-training and self-training," *Advances in Neural Information Processing Systems*, vol. 33, pp. 3833–3845, 2020.

[228] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, *et al.*, "Learning transferable visual models from natural language supervision," pp. 8748–8763, 2021.

[229] S. Min, X. Lyu, A. Holtzman, M. Artetxe, M. Lewis, H. Hajishirzi, and L. Zettlemoyer, "Rethinking the role of demonstrations: What makes in-context learning work?," *arXiv preprint arXiv:2202.12837*, 2022.

[230] J. Liu, D. Shen, Y. Zhang, B. Dolan, L. Carin, and W. Chen, "What makes good in-context examples for gpt-3?," *arXiv preprint arXiv:2101.06804*, 2022.

[231] S. Mishra, D. Khashabi, C. Baral, and H. Hajishirzi, "Reframing instructional prompts to gptk's language," *arXiv preprint arXiv:2109.07830*, 2022.

[232] A. Wettig, T. Gao, Z. Zhong, and D. Chen, "Should you mask % in masked language modeling?," *arXiv preprint arXiv:2202.08005*, 2023.

[233] A. Roy, P. Deb, P. Deb, S. Jana, A. Chakraborty, *et al.*, "Adaptive curriculum learning for genomic transformers improves model convergence," *bioRxiv*, 2024.

[234] M. Cheng, S. Wang, Y. Park, *et al.*, "Focused masking strategy for antibody complementarity determining regions in language models," *Bioinformatics*, vol. 40, no. 2, p. btad770, 2024.

[235] Y. Lin, D. Wang, W. Wu, Y. Qi, Q. Wang, and X. Zhang, "Multi-scale masked autoencoders for single-cell gene expression imputation," *BMC Bioinformatics*, vol. 25, p. 147, 2024.

[236] B. Cheng, A. Schwing, and A. Kirillov, "Per-pixel classification is not all you need for semantic segmentation," *Advances in Neural Information Processing Systems*, vol. 34, pp. 17864–17875, 2021.

[237] M. Zaheer, G. Guruganesh, K. A. Dubey, J. Ainslie, C. Alberti, S. Ontanon, P. Pham, A. Ravula, Q. Wang, L. Yang, *et al.*, "Big bird: Transformers for longer sequences," *Advances in Neural Information Processing Systems*, vol. 33, pp. 17283–17297, 2020.

[238] Y. Zhang *et al.*, "Comparisons of computation algorithms for single-cell rna-seq," *Briefings in Functional Genomics*, vol. 18, no. 6, pp. 349–356, 2019.

[239] K. Van den Berge, F. Perraudeau, C. Soneson, M. I. Love, D. Risso, J.-P. Vert, M. D. Robinson, S. Dudoit, and L. Clement, "Observation weights unlock bulk rna-seq tools for zero inflation and single-cell applications," *Genome Biology*, vol. 19, p. 24, 2018.

[240] E. Pierson and C. Yau, "Zifa: Dimensionality reduction for zero-inflated single-cell gene expression analysis," *Genome Biology*, vol. 16, p. 241, 2015.

[241] H. T. N. Tran, K. S. Ang, M. Chevrier, X. Zhang, N. Y. S. Lee, M. Goh, and J. Chen, "A benchmark of batch-effect correction methods for single-cell rna sequencing data," *Genome Biology*, vol. 21, p. 12, 2020.

[242] B. Hie, B. Bryson, and B. Berger, "Efficient integration of heterogeneous single-cell transcriptomes using scanorama," *Nature Biotechnology*, vol. 37, pp. 685–691, 2019.

[243] L. Haghverdi, A. T. Lun, M. D. Morgan, and J. C. Marioni, "Batch effects in single-cell rna-sequencing data are corrected by matching mutual nearest neighbors," *Nature Biotechnology*, vol. 36, pp. 421–427, 2018.

[244] I. Korsunsky, N. Millard, J. Fan, K. Slowikowski, F. Zhang, K. Wei, Y. Baglaenko, M. Brenner, P.-r. Loh, and S. Raychaudhuri, "Fast, sensitive and accurate integration of single-cell data with harmony," *Nature Methods*, vol. 16, pp. 1289–1296, 2019.

[245] V. R. Knight-Schrijver, H. Davaapil, S. Bayraktar, *et al.*, "A single-cell comparison of adult and fetal human epicardium defines the age-associated changes in epicardial activity," *Nature Cardiovascular Research*, vol. 1, pp. 1215–1229, 2022.

[246] G. X. Y. Zheng, J. M. Terry, P. Belgrader, P. Ryvkin, S. J. Bent, R. Wilson, S. B. Ziraldo, T. D. Wheeler, G. P. McDermott, J. Zhu, *et al.*, "Massively parallel digital transcriptional profiling of single cells," *Nature Communications*, vol. 8, p. 14049, 2017.

[247] P. D. Thomas, D. Ebert, A. Muruganujan, T. Mushayahama, L.-P. Albou, and H. Mi, "Panther: Making genome-scale phylogenetics accessible to all," *Protein Science*, vol. 31, no. 1, pp. 8–22, 2022.

[248] S. Khaldoyanidi, D. Nagorsen, A. Stein, G. Ossenkoppele, and M. Subklewe, “Immune biology of acute myeloid leukemia: Implications for immunotherapy,” *J. Clin. Oncol.*, vol. 39, pp. 419–432, Feb. 2021.

Appendices

A Partial Finetuning Results (MAM)

Table 10: Appendix: Partial-finetune results on the Hematopoietic Niche dataset. (FT = finetune, P = pretraining, Orig. = original backbone weights, all tokens = masked all attention-suggested tokens, 80/10/10 = the standard random 80/10/10 scheme, Injection: which stream(s) received cross-attention output). Here Orig. comes from [1]. Results are reported as the mean ± confidence interval from 5-fold cross-validation on the test set, with each fold held out once.

Exp	Masking	Injection	Train	F1-Score (%)
0	Random (80/10/10)	-	Orig. + FT	80.04 ± 0.001
1	Random (80/10/10)	-	P+FT	79.33 ± 0.001
2	Random (All tokens)	-	P+FT	38.80 ± 0.002
3	MAM (80/10/10)	Both	P+FT	60.40 ± 0.003
4	MAM (All tokens)	Both	P+FT	80.86 ± 0.001
5	MAM (80/10/10)	Separate	P+FT	55.19 ± 0.003
6	MAM (All tokens)	Separate	P+FT	81.31 ± 0.002
7	MAM (80/10/10)	Encoder	P+FT	70.19 ± 0.001
8	MAM (All tokens)	Encoder	P+FT	81.10 ± 0.001
9	MAM (80/10/10)	Decoder	P+FT	31.30 ± 0.002
10	MAM (All tokens)	Decoder	P+FT	81.33 ± 0.002

Table 11: Appendix: Partial-finetune results on the PBMC68K dataset. Same notation as Table 10. Here Orig. comes from [1]. Results are reported as the mean ± confidence interval from 5-fold cross-validation on the test set, with each fold held out once.

Exp	Masking	Injection	Train	F1-Score (%)
0	Random (80/10/10)	-	Orig. + FT	77.62 ± 0.004
1	Random (80/10/10)	-	P+FT	77.09 ± 0.005
2	Random (All tokens)	-	P+FT	77.06 ± 0.004
3	MAM (80/10/10)	Both	P+FT	78.71 ± 0.006
4	MAM (All tokens)	Both	P+FT	79.02 ± 0.011
5	MAM (80/10/10)	Separate	P+FT	78.13 ± 0.007
6	MAM (All tokens)	Separate	P+FT	78.33 ± 0.007
7	MAM (80/10/10)	Encoder	P+FT	78.23 ± 0.008
8	MAM (All tokens)	Encoder	P+FT	79.58 ± 0.009
9	MAM (80/10/10)	Decoder	P+FT	77.89 ± 0.006
10	MAM (All tokens)	Decoder	P+FT	78.36 ± 0.007

Table 12: Appendix: Partial-finetune results on the Heart dataset. Same notation as Table 10. Here Orig. comes from [1]. Results are reported as the mean ± confidence interval from 5-fold cross-validation on the test set, with each fold held out once.

Exp	Masking	Injection	Train	F1-Score (%)
0	Random (80/10/10)	-	Orig. + FT	91.63 ± 0.004
1	Random (80/10/10)	-	P+FT	90.90 ± 0.004
2	Random (All tokens)	-	P+FT	90.88 ± 0.005
3	MAM (80/10/10)	Both	P+FT	92.06 ± 0.008
4	MAM (All tokens)	Both	P+FT	92.37 ± 0.004
5	MAM (80/10/10)	Separate	P+FT	91.87 ± 0.005
6	MAM (All tokens)	Separate	P+FT	92.31 ± 0.005
7	MAM (80/10/10)	Encoder	P+FT	92.19 ± 0.005
8	MAM (All tokens)	Encoder	P+FT	92.34 ± 0.005
9	MAM (80/10/10)	Decoder	P+FT	92.22 ± 0.007
10	MAM (All tokens)	Decoder	P+FT	92.57 ± 0.006

B Linear Probe Results (MAM)

Here, we present the results of linear probing for Hematopoietic Niche and PBMC68K datasets, in Table 13 and Table 14, respectively.

Table 13: Appendix: Linear-probe results on the Hematopoietic Niche dataset. Same notation as Table 10. Here Orig. comes from [1]. Results are reported as the mean ± confidence interval from 5-fold cross-validation on the test set, with each fold held out once.

Exp	Masking	Injection	Train	F1-Score (%)
0	Random (80/10/10)	-	Orig. + LP	77.36 ± 0.002
1	Random (80/10/10)	-	P+LP	26.51 ± 0.002
2	Random (All tokens)	-	P+LP	28.51 ± 0.003
3	MAM (80/10/10)	Both	P+LP	36.55 ± 0.002
4	MAM (All tokens)	Both	P+LP	70.55 ± 0.001
5	MAM (80/10/10)	Separate	P+LP	29.81 ± 0.001
6	MAM (All tokens)	Separate	P+LP	70.59 ± 0.001
7	MAM (80/10/10)	Encoder	P+LP	38.15 ± 0.002
8	MAM (All tokens)	Encoder	P+LP	49.23 ± 0.002
9	MAM (80/10/10)	Decoder	P+LP	20.15 ± 0.002
10	MAM (All tokens)	Decoder	P+LP	71.54 ± 0.001

Table 14: Appendix: Linear-probe results on the PBMC68K dataset. Same notation as Table 10. Here Orig. comes from [1]. Results are reported as the mean ± confidence interval from 5-fold cross-validation on the test set, with each fold held out once.

Exp	Masking	Injection	Train	F1-Score (%)
0	Random (80/10/10)	-	Orig.+ LP	70.67 ± 0.005
1	Random (80/10/10)	-	P+LP	65.55 ± 0.010
2	Random (All tokens)	-	P+LP	65.06 ± 0.008
3	MAM (80/10/10)	Both	P+LP	68.09 ± 0.010
4	MAM (All tokens)	Both	P+LP	68.14 ± 0.010
5	MAM (80/10/10)	Separate	P+LP	68.32 ± 0.011
6	MAM (All tokens)	Separate	P+LP	68.25 ± 0.013
7	MAM (80/10/10)	Encoder	P+LP	68.13 ± 0.010
8	MAM (All tokens)	Encoder	P+LP	68.17 ± 0.010
9	MAM (80/10/10)	Decoder	P+LP	68.09 ± 0.008
10	MAM (All tokens)	Decoder	P+LP	68.43 ± 0.010

Table 15: Appendix: Linear-probe results on the Heart dataset. Same notation as Table 10. Here Orig. comes from [1]. Results are reported as the mean ± confidence interval from 5-fold cross-validation on the test set, with each fold held out once.

Exp	Masking	Injection	Train	F1-Score (%)
0	Random (80/10/10)	-	Orig.+ LP	91.56 ± 0.004
1	Random (80/10/10)	-	P+LP	89.99 ± 0.006
2	Random (All tokens)	-	P+LP	90.10 ± 0.008
3	MAM (80/10/10)	Both	P+LP	80.11 ± 0.010
4	MAM (All tokens)	Both	P+LP	89.76 ± 0.007
5	MAM (80/10/10)	Separate	P+LP	90.11 ± 0.005
6	MAM (All tokens)	Separate	P+LP	90.13 ± 0.006
7	MAM (80/10/10)	Encoder	P+LP	89.55 ± 0.005
8	MAM (All tokens)	Encoder	P+LP	89.61 ± 0.007
9	MAM (80/10/10)	Decoder	P+LP	89.24 ± 0.006
10	MAM (All tokens)	Decoder	P+LP	89.57 ± 0.008