MAM: Multinomial Attention Masking for Foundation Models on Sparse Single-Cell RNA-seq Data

Amirreza Naziri NAZIRIAM@YORKU.CA
York University, Canada
Vector Institute

Arash Asgari ARASHASG@YORKU.CA
York University
Vector Institute

Aijun An AAN@YORKU.CA
York University

Eleftherios Sachlos SACHLOS@YORKU.CA
York University

Laleh Seyyed-Kalantari LSK@YORKU.CA
York University
Vector Institute
CIFAR Solution Network

Abstract

Single-cell RNA sequencing (scRNA-seq) has transformed biology by enabling the measurement of gene expression across millions of individual cells, revealing cellular heterogeneity that underlies development, disease progression, and treatment response. This has made scRNA-seq a central data modality in modern biology and drug discovery. Recently, transformer-based foundation models (FMs) have shown strong potential for scRNA-seq analysis, but they often rely on random masking during training. Due to the extreme sparsity of scRNA-seq datasets, conventional uniform masking samples genes without considering their biological importance. In this work, we propose Multinomial Attention Masking (MAM), a module that learns which gene positions are most informative to mask at each training step. We define a set of trainable latent vectors that attend over gene embeddings to produce attention maps, from which a multinomial sampler selects the highest-scoring positions for masking. We show MAM improves FMs pretraining performance and consistently outperforms uniform masking on cell-type classification tasks, while adding negligible computational overhead. Our work benefits researchers building FMs for sparse data and those rely on accurate scRNA-seq analysis to study cell types and disease.

Data and Code Availability scRNA-seq data consist of large, sparse matrices of gene activity values across cells, ranging from 0 to large real numbers, with higher values showi higher gene expression. We used four publicly available scRNA-seq datasets spanning diverse tissues and conditions. To avoid data leakage, we verified that *none of these datasets* were included in the backbone model's Hao et al. (2024) pretraining data.

The **Hematopoietic Niche** dataset Ennis et al. (2023) profiles 300K cells from 50 donors, capturing heterogeneous transcriptional programs across leukemia states. **PBMC68K** Zheng et al. (2017) contains 68K peripheral blood mononuclear cells from healthy donors and serves as a benchmark for reconstruction, representation learning, and clustering. The **Heart** dataset Knight-Schrijver et al. (2022) includes 60K single-cell profiles from human cardiac tissue, covering major cell types and tissue heterogeneity. **Retina** Wang et al. (2022) is a joint atlas of gene expression and chromatin accessibility from 50K+ adult human retinal cells, supporting studies of gene regulation and ocular disease.

We released our public code repository here. [1]

1. https://github.com/Amir79Naziri/MAM_code

Institutional Review Board (IRB) This research use exiting public data, and does not require IRB approval.

1. Introduction

Single-cell RNA sequencing (scRNA-seq) has revolutionized our understanding of cellular heterogeneity by measuring transcriptomes at the resolution of individual cells Kolodziejczyk et al. (2015); Saliba et al. (2014). scRNA-seq has enabled the discovery and annotation of novel cell types, inference of gene regulatory networks, identification of rare subpopulations, and profiling of cell-specific responses to drugs and perturbations Van de Sande et al. (2023); Jovic et al. (2022); Hedlund and Deng (2018).

In parallel, Transformer-based foundation models (FMs) are large neural networks pretrained on massive unlabeled data and finetuned for diverse tasks. Originally developed for NLP (e.g., BERT Devlin et al. (2019), GPT-3 Brown et al. (2020)) and vision (e.g., MAE He et al. (2022)), they typically consist of three core modules Patwardhan et al. (2023); Han et al. (2023): (a) token (and positional) embedding layers map discrete input units into continuous vectors; (b) encoder layers aggregate and transform embeddings into high-level representations; and (c) decoder layers reconstruct the original inputs (or predict targets) from those representations. FMs are typically trained with two self-supervised objectives: masked language modeling, which reconstructs a small fraction of masked tokens to learn bidirectional context, and autoregressive modeling, which predicts tokens sequentially to model the joint distribution and supports generation Jaiswal et al. (2020).

Despite the success of masked and autoregressive language modelling in natural language and vision domains, applying these paradigms to scRNA-seq data remains challenging. scRNA-seq data are characterized by extreme sparsity (dropout), high dimensionality (tens of thousands of genes), and substantial technical noise Zhang et al. (2025); Imoto et al. (2022), which complicate tasks such as cell-type classification Erfanian et al. (2023). In masked language modelling, uniform random masking often selects non-informative genes with limited biological signal. As a result, most masked positions correspond to dropout events and non-variable genes rather than meaningful variation, thereby reducing the model's learning efficiency in this domain. Consequently, naively applying random masking when adapting new FMs can not only fail to improve performance but may even degrade it. In addition, because masking is random, it is not possible to study masked genes or the masking effects on the different modules of FMs.

Recently, self-guided Masked Autoencoders (SMA) Xie et al. (2024) were introduced that learn domain-agnostic masking patterns from attention maps, which could reveal hidden structures in scRNA FMs, and inform strategies to improve model performance. However, SMA relies on a teacher–student setup with high computational cost. On scRNA-seq, a single cell's 20 k-token sequence already exceeds typical GPU memory budgets (Appendix Table 6). SMA was tested on dense, low-dimensional data and does not address the extreme zero–nonzero imbalance in single-cell gene expression, nor does it allow analysis of how masking affects different model components.

These limitations suggest that conventional random masking strategies are ill-suited for scRNA-seq data, as they fail to prioritize biologically informative genes and are dominated by dropout-driven zero entries. Also, the implementation does not allow us to investigate how masking information is utilized in different parts of the model. In contrast, self-guided masking can direct the model's attention toward important genes and meaningful expression patterns. This leaves us with this question:

Can we propose a lightweight and interpretable attention-guided masking strategy that focuses on biologically important genes, and improves foundation model pretraining on sparse scRNA-seq data?

In this work, we propose a novel Multinomial Attention Masking (MAM) approach, tailored to the unique challenges of scRNA-seq data. Our method can be summarized in three steps: (a) At each pretraining step, latent vectors attend over gene embeddings, producing attention scores that identify the most informative genes. (b) A multinomial sampling procedure uses these scores to select positions to mask, focusing reconstruction on biologically meaningful variation. (c) The latent summary is then injected back into the network via one of four context-injection schemes (encoder-only, decoder-only, shared adapter, separate adapters), enabling efficient, robust scaling of FMs on scRNA-seq data. As a result, MAM ensures that masking prioritizes genuinely informative, variable gene expressions. By combining this with lightweight context-injection adapters, MAM remains computationally efficient and, more importantly, lets us test how masking signals prop-

agate through different FM modules. Our contributions are as follows:

- We propose MAM, a dynamic and efficient attention-guided masking method for sparse, noisy data, and compare it against uniform random masking.
- We introduce a context-injection scheme in MAM and evaluate masking across four FM settings on four datasets (encoder-only, decoder-only, both, and both with separate adapters), showing adaptability across architectures.
- We show MAM outperforms random masking in training and improves cell type classification performance.
- We use biological insights to show MAM's ability to select important genes.

2. Related works

Foundation models for single-cell RNA data: Recent work has begun to translate FMs paradigm into transcriptomics. For example, scBERT Yang et al. (2022), an encoder-only FM, adapts the BERT masked-language framework to gene expression by learning contextualized embeddings of genes and cells through token-level reconstruction on discretized counts. scGPT Cui et al. (2024), a decoder-only FM, learns joint embeddings of cells and genes via masked language modeling on multi-omic single-cell data; xTrimoGene Gong et al. (2023) exploits sparse cross-attention to process all 20,000 genes without manual discretization, completing that, scFoundation Hao et al. (2024) scales to 100 M parameters by incorporating read-depth–aware objectives over 50 M cell profiles. The core idea in both xTrimoGene and scFoundation is skipping zero and masked tokens in encoder layers, which can significantly reduce the computing needs without losing performance. To the best of our knowledge, scFoundation is the only encoder-decoder FM model in this area. All of these models use random masking, thus it is not possible to investigate masked tokens and interpret how masking information is utilized throughout the layers.

Autoguided masking in self-supervision: The paper by Xie et al. (2024) introduces Self-guided Masked Autoencoders (SMA) across image, text, chemical-graph, and particle-physics modalities, using the model's own early attention maps to steer mask selection. However, SMA relies on a complex teacher–student framework and was evaluated on relatively dense, low-dimensional inputs; it does not address the severe zero/non-zero imbalance in scRNA-seq, where most masks still fall on uninformative zeros. The ADIOS approach Shi et al. (2022) further couples an adversarially trained masking network with a Siamese encoder to produce maximally challenging masks, yielding strong gains on vision benchmarks. Yet its adversarial objective and multi-mask sampling incur substantial complexity and do not target the sparsity characteristics of single-cell data. Within the single-cell domain, Fang et al. Fang et al. (2024) propose scMAE, which perturbs gene expression via controlled shuffling and employs a linear "mask predictor" after the encoder to guide reconstruction. However, scMAE still samples masks uniformly at random and cannot be classified as an autoguided masking strategy.

To our knowledge, none of these self-guided masking methods are explicitly designed for the high-dimensional, noise-prone sparsity of scRNA-seq. In contrast, our masking method, MAM, offers a lightweight, attention-driven mechanism that focuses on informative genes in single-cell transcriptomes.

3. Method

We introduce a novel masking method, MAM, that can be integrated into off-the-shelf FMs. The method learns feature importance via learnable parameters and incorporates a unit with a cross-attention context injection to better handle sparse scRNA-seq data.

3.1. Latent Cross-Attention Module

We consider an input dataset of single-cell gene expression profiles. 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. 1. Formally, let:

$$Z = \left[z_1, z_2, \ldots, z_N\right]^\top \in \mathbb{R}^{N \times D_z}, \quad z_n \in \mathbb{R}^{D_z} \tag{1}$$

We apply multi-head cross-attention with H heads, using Z as the set of queries and X_{proj} as keys and values. To match the shape requirements of standard multi-head attention implementations, we reshape:

$$Q = Z \in \mathbb{R}^{N \times B \times D_z}, \quad K = V = X_{proj} \in \mathbb{R}^{L \times B \times D_z}. \tag{2}$$

The cross-attention then produces:

$$\big(O,\ A\big) = \mathrm{CrossAttn}(Q, K, V), \tag{3}$$

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).

3.2. 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:

$$S_{b,n,t} = \sum_{h=1}^{H} A_{b,h,n,l}, \quad \textit{where} \quad S \in \mathbb{R}^{B \times N \times L}. \tag{4}$$

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 Xie et al. (2024), attention maps often exhibit heavy overlap. Many queries focus on highly similar or correlated token groups. This hinders the construction of diverse, informative masking patterns when all heads or latents are used, leading to concentrated and redundant masks. We aggregate the selected latents' scores by summing to obtain token-level importance.

$$s_{b,l} = \sum_{n \in R} S_{b,n,l}, \quad s \in \mathbb{R}^{B \times L} \tag{5}$$

Matrix s measures each token's aggregate attention over the selected latents. We add a small constant $\varepsilon > 0$ (division-by-zero avoidance) and normalize:

$$p_{b,l} = \frac{s_{b,l} + \varepsilon}{\sum_{l'=1}^{L} \big(s_{b,l'} + \varepsilon\big)}, \quad \textit{where} \quad \sum_{t=1}^{L} p_{b,l} = 1. \tag{6}$$

Finally, for each cell b, we sample $k = \lfloor \rho \times L \rfloor$ distinct token indices via multinomial sampling from $p_{b,1}, \ldots, p_{b,L}$, where ρ is the masking ratio. These k tokens are masked, and all others remain visible to both the encoder and decoder.

3.3. Cross-Attention Context Injection

To enable the latent vectors learning which token features are most informative and explore masking information throughout the modules, we inject the cross-attention outputs from Eq. 3 back into the backbone network. First, we pool O over the N latents:

$$o_b \ = \ \frac{1}{N} \sum_{n=1}^{N} O_{n,b,:} \ \in \ \mathbb{R}^{D_z}, \quad b = 1, \ldots, B, \tag{7}$$

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:

$$\begin{aligned} \mathrm{adapter}(o_b) \ &= \ W_{\mathrm{adapt}}\, o_b \ + \ b_{\mathrm{adapt}}, \\ W_{\mathrm{adapt}} \in \mathbb{R}^{D_e \times D_z}; \quad & b_{\mathrm{adapt}} \in \mathbb{R}^{D_e}. \end{aligned} \tag{8}$$

Here, W_{adapt} and b_{adapt} are learnable parameters. The resulting vector is broadcasted to every time step of the chosen stream(s).

There are four different ways (Fig. 1) to perform context injection, as follows:

(A) **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.

(B) **Decoder-only**: We inject the attention output into the decoder, enabling the latents to capture reconstruction dynamics and identify regions that are harder to reconstruct.

(C) **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.

(D) **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. Self-supervised Pretraining

We train the backbone separately on each dataset (see Section **Experiments**). For all datasets, we apply a masking ratio, ρ, of 30% during pretraining, same as the backbone model default setup Hao et al. (2024). 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 ρ and follows the same standard strategy as the backbone model.

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:

$$\mathcal{L} = \frac{1}{|\mathcal{M}|} \sum_{(b,t)\in\mathcal{M}} (\hat{x}_{b,t} - x_{b,t})^2 \; + \; \lambda_{\text{attn}} \|O\|_2^2, \qquad \lambda_{\text{attn}} = 10^{-5}. \tag{9}$$

The reconstruction term forces the decoder to recover masked gene expression values, while the L_2 regularization on the cross-attention outputs O serves a dual purpose: it prevents activations from growing without bound (avoiding representation collapse observed in random masking) and encourages smoother, less peaky attention distributions. This mathematical constraint ensures that MAM maintains high-quality representations even when masking "all tokens," a regime that typically destabilizes uniform random baselines.

3.5. Model

We evaluate our method on scFoundation Hao et al. (2024), an encoder-decoder FM that allows us to simulate encoder-only and decoder-only settings within a single backbone. This choice avoids the high computational cost of pretraining multiple large models while demonstrating that our approach is compatible with different model architectures. Other popular models, such as scBERT and scGPT, do not provide either pretrained weights or pretraining code, making large-scale retraining infeasible under our resources. Since all these models rely on uniform random masking during pretraining, using scFoundation provides a fair and practical testbed. Finally, we ensure that all downstream datasets are disjoint from the scFoundation pretraining corpus.

3.6. Training and Validation Setup

To validate our MAM algorithm, we employ a two-stage training protocol: self-supervised pretraining and supervised finetuning. In addition, for completeness, we summarize the total compute resources and runtime across all experiments in Appendix Table 6.

Self-supervised Pretraining Setup: 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 Hao et al. (2024) 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.

Moreover, we explore four distinct context-injection strategies. 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.

We train the parameters using Adam with two learning-rate groups: $\text{LR}_{\text{new}} = 10^{-4}$ for layers introduced by our method and $\text{LR}_{\text{backbone}} = 10^{-5}$ 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 Hao et al. (2024)). For more details, please refer to Table 3 in the Appendix.

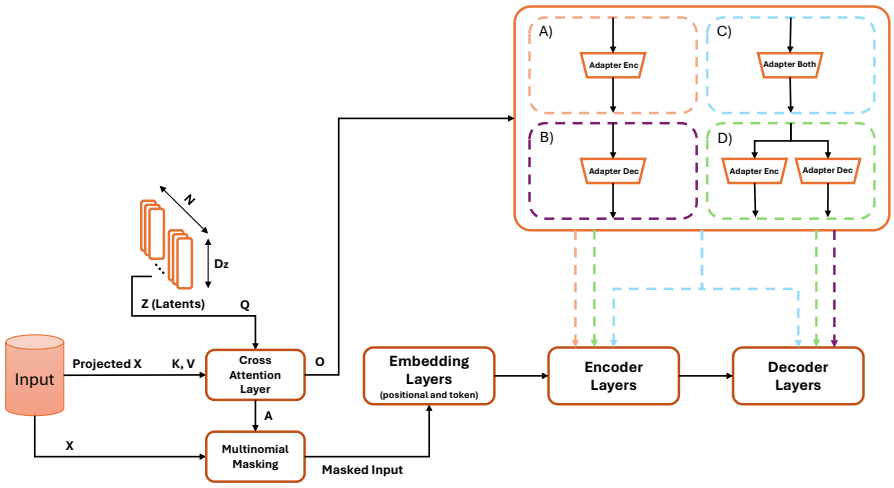

Figure 1: Proposed attention-driven masking and context-injection workflow. Learned latent vectors attend to token embeddings via cross-attention (Q, K, V) to produce attention weights A and pooled context O. A multinomial sampling of A yields the mask positions. The masked sequence is passed through 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.

Supervised Finetuning Setup: A lightweight classification head is added on top of the pretrained encoder and can be trained on different downstream tasks. Here, we chose cell type classification as it is an important task in bioinformatics, and the dataset supports it. We finetune the classification of cell types on each dataset. We append a two-layer MLP on top of the pooled encoder output. Also, we freeze all latents, cross-attention weights, and all adapter layers during finetuning, as they are only responsible for self-supervised training. We use the cross-entropy loss function, Adam optimizer with a single learning rate of 1×10^{-4}, weight decay of 5×10^{-4}, and clip gradients to the maximum norm of 1.0. Training is carried out with accumulating gradients over 5 steps. Models are trained with early stopping with a patience of 5 epochs based on validation loss (following the default finetuning hyperparameters reported for backbone Hao et al. (2024)). For more details, please refer to Table 4 in the Appendix.

We evaluate partial finetune and Linear probing schemes (on top of the classification head) to probe the usefulness of the pretrained encoder. We skipped full-finetuning due to the extensive computation resources the model required. In partial finetune, trainable parameters are: token embeddings, positional embeddings, the *last* transformer block of the encoder, and the classification head. The frozen parts of the model are all earlier encoder layers. On the other hand, for linear probing, the trainable parameters are only the classification head, and the frozen parts are token embeddings, positional embeddings, and *all* encoder layers. It is worth mentioning that all masking and cross-attention modules were frozen during finetuning. This is because all those modules are solely added to make the pretraining more robust. If we do not freeze them in the finetuning step, it will be similar to adding extra parameters to the classification head, which is not our goal.

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 Thomas et al. (2022), which identifies the biological processes of a set of genes. Using GO en-

Table 1: Partial-finetune cell-type classification, comparing MAM (Ours) F1-score vs random masking. (FT: finetune, P: pretraining, Orig. = original backbone weights Hao et al. (2024), 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 mean ± confidence interval over 5-fold cross-validation, with each fold held out once for testing.

Dataset	Masking	Injection	Train	F1-score
Hematopoietic	Random (80/10/10)	–	Orig.Hao et al. (2024) + 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.Hao et al. (2024) + 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.Hao et al. (2024) + 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**
Retina	Random (80/10/10)	–	Orig.Hao et al. (2024) + FT	98.93 ± 0.002
	Random (80/10/10)	–	P+FT	98.33 ± 0.002
	MAM (all tokens)	Both	P+FT	**99.44 ± 0.002**
	MAM (all tokens)	Separate	P+FT	98.81 ± 0.002
	MAM (all tokens)	Encoder	P+FT	98.44 ± 0.002
	MAM (all tokens)	Decoder	P+FT	98.92 ± 0.003

richment, we can determine if the masked gene is informative or not.

4. Results

Here, due to space constraints, only a subset of cell-type classification results is shown in the main text; full results are provided in Appendix Tables 7–13.

4.1. MAM outperforms random masking under partial finetuning

Tables 1 compare MAM against the standard uniform random 80/10/10 masking under both partial finetuning (FT). Here, P, FT denote pretraining and finetuning, and pretraining with finetuning respectively. Here, MAM consistently outperforms random masking on all four datasets. On Hematopoietic Niche, random masking reaches 79.33% (vs. 80.04% for Orig.+FT), while MAM achieves up to 81.33% (decoder injection), a +2.00% absolute gain over random and +1.29% over Orig.+FT. On PBMC68K, random masking reaches 77.09% (Orig.+FT: 77.62%), whereas MAM reaches 79.58% (encoder injection), a +2.49% gain over random (+1.96% over Orig.+FT). We observe similar improvements on Heart (90.90% → 92.57%, +1.67%) and Retina (98.33% → 99.44%, +1.11%), indicating that attention-guided masking yields more transferable representations than uniform random masking under finetuning.

4.2. MAM in linear probing vs fine tuning

Under P+LP (Table 2), random masking collapses on the Hematopoietic Niche dataset (77.36% → 26.51%), indicating that uniform masking during pretraining can substantially distort representations that were previously well structured for linear separation. Because the backbone remains frozen during LP, such degradation cannot be corrected by the shallow clas-

Table 2: Linear-probe (LP) cell-type classification (F1-score). Same notation as Table 1. On large datasets (Hematopoietic), and difficult one with lower performance (PBMC68K), random masking with P+LP underperforms because a shallow classifier cannot compensate for suboptimal pretrained representations; MAM recovers performance by learning more task-relevant features. On smaller, easier datasets (Heart, Retina), FM performance is high, and LP is sufficient to fit the data.

Dataset	Masking	Injection	Train	F1-score
Hematopoietic	Random (80/10/10)	–	Orig.Hao et al. (2024) + 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.Hao et al. (2024) + 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.Hao et al. (2024) + LP	**91.56 ± 0.004**
	Random (80/10/10)	–	P+LP	89.99 ± 0.006
	MAM (all tokens)	Both	P+LP	89.76 ± 0.007
	MAM (all tokens)	Separate	P+LP	90.13 ± 0.006
	MAM (all tokens)	Encoder	P+LP	89.61 ± 0.007
	MAM (all tokens)	Decoder	P+LP	89.57 ± 0.008
Retina	Random (80/10/10)	–	Orig.Hao et al. (2024) + LP	84.03 ± 0.003
	Random (80/10/10)	–	P+LP	**97.86 ± 0.004**
	MAM (all tokens)	Both	P+LP	96.80 ± 0.001
	MAM (all tokens)	Separate	P+LP	95.15 ± 0.004
	MAM (all tokens)	Encoder	P+LP	94.18 ± 0.002
	MAM (all tokens)	Decoder	P+LP	95.45 ± 0.004

sifier alone, particularly in a large dataset such as Hematopoietic Niche, as there are not enough parameters to learn the pattern in a shallow classifier. In contrast, MAM alleviates this issue by guiding pretraining toward more task-relevant gene representations, recovering performance to 71.54% with decoder injection and closing most of the gap to the original backbone. These results are not recoverable in the encoder-only model as the encoder is also frozen in LP. On PBMC68K, MAM also provides consistent, though more modest, improvements over random masking. In Heart, and Retina, the datasets size is smaller than Hematopoietic Niche and the tasks are easier compare to PBMC68K, therefore, a shallow classifier alone can still learn the data pattern.

The different behavior between LP and partial FT reflects their fundamentally different adaptation capacities. LP restricts learning to the final classifier, thereby exposing any mismatch between the pretraining objective and the downstream task. Partial FT, in contrast, allows the encoder to update and re-align internal representations, mitigating the negative effects introduced during pretraining. Thus, in partial FT, MAM consistently outperforms random masking across all four datasets and in several cases even surpasses the original pretrained backbone.

4.3. The impact of context-injection schemes

Context-injection in general is critical for MAM success. Under partial finetuning (Table 1), the location where the learned latent parameters are injected is not sensitive across Encoder, Decoder, or Both schematic and under all those cases, MAM often outperform. We observe that the optimal strategy is not consistent, but the differences in performance are small. Therefore, the location of injecting the latent parameters is not a critical design choice. Under linear probing (Table 2), when the backbone is frozen, including the encoder, then the decoder and

both-stream injections recover most of the original backbone's performance on Hematopoietic Niche and PBMC68K. However, encoder-only injection is less effective, as the encoder is kept frozen in LP. on Heart and Retina, where the tasks are initially easier for the model, the differences between the locations of context-injections are smaller and random masking remains competitive. Overall, these results show that the location of context-injection is not a key design choice in MAM, but targeted injection in general provides the greatest benefit, especially when model adaptation capacity is limited.

4.4. Masking style (80/10/10 vs. all tokens)

Here, we compare the standard 80/10/10 masking schedule with masking all tokens. Results are shown in Tables 7 through 13 in the Appendix.

For uniform random masking, switching to all tokens for the Hematopoietic Niche under partial finetuning, is catastrophic, dropping performance from 79.33% to 38.80%. On PBMC68K and Heart, the effect is negligible, from 77.09% to 77.06% and from 90.90% to 90.88%. On Retina, it slightly improves performance from 98.33% to 98.47%. Under linear probing, random all token masking leads to marginal changes, for example from 26.51% to 28.51% on the niche and from 65.55% to 65.06% on PBMC68K. Overall, randomly masking all tokens is unstable and does not consistently improve performance.

In contrast, within MAM, masking all tokens is generally beneficial and often necessary for strong results. Across datasets and evaluation modes, the all token variant consistently matches or improves over the 80/10/10 MAM counterpart, with especially large gains on the Hematopoietic Niche under both partial finetuning and linear probing.

These results indicate that masking all tokens is risky under uniform random masking, particularly in sparse settings such as the niche, but is well supported within MAM where masking is guided by attention. Detailed comparisons are provided in Tables 7 through 13 and the corresponding Heart and Retina tables in the Appendix.

4.5. Analyzing Learned Masked Tokens

In this part, we want to explore if the Masked Autoencoder for scRNA-seq, MAM, learn to selectively mask biologically informative genes, rather than masking features randomly. Therefore, we examined, gene-by-gene, how often each feature was chosen for masking. Specifically, for every gene in the training set, we (i) count the total number of masking events across all iterations and (ii) compute that gene's mean expression level conditional on being masked. Genes with higher expression levels often carry more biological information. If these genes are more frequently masked, it's a sign that the model may be learning to focus on informative features. On the other hand, if masking were uniform/random, all genes would be masked with roughly equal frequency over time, indicating that masking is random.

Plotting gene's masking frequency against mean expression when masked (see Fig. 2) reveals a positive correlation: genes that register non-zero expression, and especially those with higher expression magnitudes, are selected for masking far more frequently than genes that sit at or near zero. This pattern is exactly what we would expect if the model's attention-driven sampling policy is successfully identifying the most information-rich coordinates of the input profile. Conversely, genes that are silent or expressed only sporadically supply little discriminative signal; masking them would waste capacity without providing useful learning signals. Therefore, MAM effectively masks biologically informative genes. Further architectural analysis confirming that MAM dynamically pivots focus based on individual cell-type profiles—rather than acting as a static global filter, is provided in Appendix.

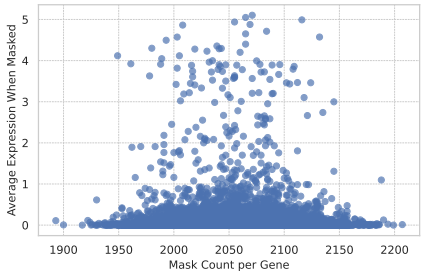

Figure 2: Scatter plot of each gene's masking frequency against its mean expression when masked (experiment 8, Appendix Table 8), illustrating that MAM preferentially targets genes with higher expression levels.

4.6. Biological meaning of the masked genes

To understand why MAM improves performance, we analyze the genes most frequently selected for mask-

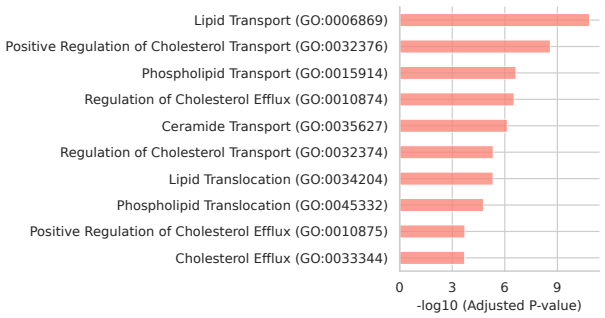

Figure 3: Top enriched GO biological processes among the most frequently masked genes in the PBMC68K dataset. The x-axis show the significance of each biological process (y-axis) derived from masked genes.

ing on the PBMC68K dataset. Gene Ontology enrichment Thomas et al. (2022) shows that these genes are strongly associated with immune related processes such as lipid transport, membrane remodeling, antigen presentation, and cytokine signaling, all of which are central to immune cell identity. By preferentially masking biologically meaningful genes (see Fig. 3), MAM encourages the model to reconstruct informative signals rather than noise, leading to richer and more discriminative representations. In contrast, uniform random masking treats all genes equally and often focuses learning on uninformative targets. These results demonstrate that attention guided masking does not merely reflect biological structure but actively helps the model capture it, improving generalization in sparse and heterogeneous single cell data. Consistent biological relevance was also observed in the Hematopoietic Niche dataset, where masked genes were significantly enriched for hematopoietic lineage pathways (see Appendix E for details).

5. Discussion and Conclusion

We introduced MAM, a novel lightweight attention-guided module that improves self-supervised pretraining of FMs on sparse scRNA-seq data by prioritizing informative genes during masking. Across four datasets, MAM consistently improves downstream cell-type classification performance and several consistent trends emerge from our experiments. First, partial finetuning (P+FT) yields stable and higher downstream performance, while adding negligible computational overhead. Second, for the linear probing protocols (P+LP), we have shown MAM is more effective when the task is hard, or the dataset is large, such that the limited parameters of the linear classifier are not adequate to learn the patterns. In (P+FT), updating the encoder enables representations to re-align after pretraining, where the encoder is kept frozen in (L+FT). Linear probing, while useful for probing representation quality, restricts adaptation to a shallow classifier and therefore reflects the compatibility between pretraining objectives and downstream tasks. Third, we found that MAM improves robustness under both evaluation protocols, with particularly strong gains observed when model adaptation capacity is limited. Fourth, among the proposed context-injection strategies, decoder-only and both-stream injection provide the most consistent improvements across datasets, while encoder-only injection is less reliable. In practice, decoder-only injection offers a strong default choice, balancing stability, performance, and simplicity.

While count-aware likelihoods like Negative Binomial (NB) loss are theoretically ideal for scRNA-seq data, we opted for MSE to maintain consistency with existing FMs (e.g., scGPT, Geneformer) and to leverage its superior numerical stability during scaled pretraining. Exploring NB-loss within the MAM framework remains a subject for future work.

Overall, MAM offers a principled alternative to uniform random masking for sparse scRNA-seq pretraining, enhancing representation transfer while retaining the strengths of large FMs. It yields more reliable models for downstream tasks, enabling sharper cell-state atlases, clearer developmental trajectories, and more precise identification of disease-relevant subpopulations and therapeutic targets.

Limitations: Our results rely on an existing, well-trained FM. A poorly pretrained backbone or domain-mismatch between pretraining and the downstream task impacts the MAM performance. Regarding computational overhead, while standard cross-attention scales quadratically, our implementation integrates Reversible Transformers and a Performer module (utilizing fast attention via orthogonal random features). This shifts the complexity boundary to linear-time $O(l)$, ensuring that the memory footprint remains manageable even for long transcriptomic sequences of 20,000+ genes.

Future work can explore more memory-efficient attention mechanisms, such as sparse or low-rank cross-attention, to further reduce computational overhead. Another promising direction is integrating MAM into large-scale pretraining from scratch, rather than treating it as a continuation of pretraining, to evaluate its impact on representation learning at scale. Extending MAM to multi-omic FMs and cross-dataset pretraining scenarios may further improve robustness to domain shift. Finally, adaptive strategies for dynamically adjusting the number of latent heads or masking ratios based on dataset sparsity could enable more efficient and scalable deployment of MAM in large single-cell atlases.

Acknowledgments

This work was supported in part by the Connected Minds Program through the Canada First Research Excellence Fund Grant #CFREF-2022-00010. We gratefully acknowledge the Vector Institute for providing high-performance computing resources. Additional support was provided by the Natural Sciences and Engineering Research Council of Canada (NSERC) through a Discovery Grant awarded to Dr. Laleh Seyyed-Kalantari, as well as by the Google Research Scholar Award and the Canadian AI Safety Institute Research Program at Canadian Institute for Advanced Research (CIFAR).

References

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020. URL https://arxiv.org/abs/2005.14165.

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

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019. URL https://arxiv.org/abs/1810.04805.

Sarah Ennis, Alessandra Conforte, Eimear O'Reilly, Javid Sabour Takanlu, Tatiana Cichocka, Sukhraj Pal Dhami, Pamela Nicholson, Philippe Krebs, Pilib Ó Broin, and Eva Szegezdi. Cell-cell interactome of the hematopoietic niche and its changes in acute myeloid leukemia. *Iscience*, 26 (6), 2023.

Nafiseh Erfanian, A Ali Heydari, Adib Miraki Feriz, Pablo Iañez, Afshin Derakhshani, Mohammad Ghasemigol, Mohsen Farahpour, Seyyed Mohammad Razavi, Saeed Nasseri, Hossein Safarpour, et al. Deep learning applications in single-cell genomics and transcriptomics data analysis. *Biomedicine & Pharmacotherapy*, 165:115077, 2023.

Zhaoyu Fang, Ruiqing Zheng, and Min Li. scmae: a masked autoencoder for single-cell rna-seq clustering. *Bioinformatics*, 40(1):btae020, 01 2024. ISSN 1367-4811. doi: 10.1093/bioinformatics/btae020. URL https://doi.org/10.1093/bioinformatics/btae020.

Jing Gong, Minsheng Hao, Xingyi Cheng, Xin Zeng, Chiming Liu, Jianzhu Ma, Xuegong Zhang, Taifeng Wang, and Le Song. xtrimogene: an efficient and scalable representation learner for single-cell rna-seq data. *Advances in Neural Information Processing Systems*, 36:69391–69403, 2023.

Kai Han, Yunhe Wang, Hanting Chen, Xinghao Chen, Jianyuan Guo, Zhenhua Liu, Yehui Tang, An Xiao, Chunjing Xu, Yixing Xu, Zhaohui Yang, Yiman Zhang, and Dacheng Tao. A survey on vision transformer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(1):87–110, 2023. doi: 10.1109/TPAMI.2022.3152247.

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

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings*

of the IEEE/CVF conference on computer vision and pattern recognition, pages 16000–16009, 2022.

Eva Hedlund and Qiaolin Deng. Single-cell rna sequencing: technical advancements and biological applications. *Molecular aspects of medicine*, 59:36–46, 2018.

Yusuke Imoto, Tomonori Nakamura, Emerson G Escolar, Michio Yoshiwaki, Yoji Kojima, Yukihiro Yabuta, Yoshitaka Katou, Takuya Yamamoto, Yasuaki Hiraoka, and Mitinori Saitou. Resolution of the curse of dimensionality in single-cell rna sequencing data analysis. *Life Science Alliance*, 5 (12), 2022.

Ashish Jaiswal, Ashwin Ramesh Babu, Mohammad Zaki Zadeh, Debapriya Banerjee, and Fillia Makedon. A survey on contrastive self-supervised learning. *Technologies*, 9(1):2, 2020.

Dragomirka Jovic, Xue Liang, Hua Zeng, Lin Lin, Fengping Xu, and Yonglun Luo. Single-cell rna sequencing technologies and applications: A brief overview. *Clinical and translational medicine*, 12 (3):e694, 2022.

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*, 1:1215–1229, 2022. doi: 10.1038/s44161-022-00183-w. URL https://doi.org/10.1038/s44161-022-00183-w.

Aleksandra A Kolodziejczyk, Jong Kyoung Kim, Valentine Svensson, John C Marioni, and Sarah A Teichmann. The technology and biology of single-cell rna sequencing. *Molecular cell*, 58(4):610–620, 2015.

Narendra Patwardhan, Stefano Marrone, and Carlo Sansone. Transformers in the real world: A survey on nlp applications. *Information*, 14(4):242, 2023.

Antoine-Emmanuel Saliba, Alexander J. Westermann, Stanislaw A. Gorski, and Jörg Vogel. Single-cell rna-seq: advances and future challenges. *Nucleic Acids Research*, 42(14):8845–8860, 07 2014. ISSN 0305-1048. doi: 10.1093/nar/gku555. URL https://doi.org/10.1093/nar/gku555.

Yuge Shi, N. Siddharth, Philip H. S. Torr, and Adam R. Kosiorek. Adversarial masking for self-supervised learning, 2022. URL https://arxiv.org/abs/2201.13100.

Paul D. Thomas, Dustin Ebert, Anushya Muruganujan, Tremayne Mushayahama, Laurent-Philippe Albou, and Huaiyu Mi. Panther: Making genome-scale phylogenetics accessible to all. *Protein Science*, 31(1):8–22, 2022. doi: https://doi.org/10.1002/pro.4218. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/pro.4218.

Bram Van de Sande, Joon Sang Lee, Euphemia Mutasa-Gottgens, Bart Naughton, Wendi Bacon, Jonathan Manning, Yong Wang, Jack Pollard, Melissa Mendez, Jon Hill, et al. Applications of single-cell rna sequencing in drug discovery and development. *Nature Reviews Drug Discovery*, 22(6):496–520, 2023.

Sean K. Wang, Surag Nair, Rui Li, Katerina Kraft, Anusri Pampari, Aman Patel, Joyce B. Kang, Christy Luong, Anshul Kundaje, and Howard Y. Chang. Single-cell multiome of the human retina and deep learning nominate causal variants in complex eye diseases. *Cell Genomics*, 2(8):100164, 2022. ISSN 2666-979X. doi: https://doi.org/10.1016/j.xgen.2022.100164. URL https://www.sciencedirect.com/science/article/pii/S2666979X22001069.

Johnathan Xie, Yoonho Lee, Annie S. Chen, and Chelsea Finn. Self-guided masked autoencoders for domain-agnostic self-supervised learning, 2024. URL https://arxiv.org/abs/2402.14789.

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

Yi Zhang, Yin Wang, Xinyuan Liu, and Xi Feng. Pbimpute: Precise zero discrimination and balanced imputation in single-cell rna sequencing data. *Journal of Chemical Information and Modeling*, 65(5):2670–2684, 2025. doi: 10.1021/acs.jcim.4c02125. URL https://doi.org/10.1021/acs.jcim.4c02125. PMID: 39957720.

Grace X. Y. Zheng, John M. Terry, Phillip Belgrader, Paul Ryvkin, Scott J. Bent, Ryan Wil-

son, Stefano B. Ziraldo, Thomas D. Wheeler, Geoffrey P. McDermott, Junjie Zhu, et al. Massively parallel digital transcriptional profiling of single cells. *Nature Communications*, 8:14049, 2017. doi: 10.1038/ncomms14049.

Appendix A. Data Preparation

We perform standard quality control on the single-cell dataset to ensure that only high-quality cells and informative genes are retained. Low-complexity cells (those with very few detected genes) and genes expressed in only a handful of cells are filtered out. Next, we partition the cleaned data into training, validation, and test sets (train: 80%, validation: 10%, test: 10%) in a way that avoids leaking information across splits. Rather than splitting at the level of individual cells, we split by donor—so that all cells from a given individual end up in the same subset. To maintain balance, we stratify the donors by their predominant combinations of cell type and timepoint, thereby ensuring that each subset reflects the overall diversity of the study. All details and codes to replicate data loading and processing are available in the provided code.

Appendix B. Training Hyper-parameters

Here, Table 3 and Table 4 present the hyperparameters we have used for self-supervised pretraining and for supervised finetuning, respectively.

Table 3: Hyperparameters for self-supervised pretraining.

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		

Appendix C. Computing Resources

We have used parallel 4 × NVIDIA A40 for computing each experiment, where each GPU has 48 GB capacity. Here, Table 6 presents the total runtime for training and validation for all the pretraining experiments.

Table 4: Hyperparameters for supervised finetuning.

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

Appendix D. Analysis of Dynamic Masking

To investigate whether MAM functions as a static global filter or a dynamic module, we analyzed the score generation process. Because the importance scores $s_{b,l}$ are derived from cross-attention between learnable latents Z and the specific projected embeddings of each input cell X_{proj}, the resulting masking distribution is mathematically dependent on the individual transcriptomic profile. This ensures that the model dynamically shifts focus, for example, masking specific lineage markers only when those features are present in the input, rather than simply targeting a fixed set of high-variance genes across the entire dataset.

Appendix E. Additional Biological Validation: Hematopoietic Niche

To demonstrate that MAM consistently captures biologically relevant signals across different tissues, we performed Gene Ontology (GO) enrichment analysis on the most frequently masked genes from the Hematopoietic Niche dataset. As shown in Table 5, the identified pathways are highly relevant to the hematopoietic lineage and immune regulation. This confirms that MAM's attention mechanism successfully prioritizes genes integral to cell-specific identity across diverse biological contexts.

Table 5: Enriched GO Biological Processes for frequently masked genes in the Hematopoietic Niche dataset.

GO ID	Biological Process
GO:0002250	Adaptive immune response
GO:0042110	T cell activation
GO:0006954	Inflammatory response
GO:0045087	Innate immune response
GO:0001816	Cytokine production

Appendix F. Partial Finetuning Results

Here, we present the full results of partial finetuning for cell-type classification across all four datasets. We report the F1-scores for Hematopoietic Niche, PBMC68K, Heart, and Retina in Table 7, Table 8, Table 9, and Table 10, respectively. It is worth mentioning that we compared our proposed MAM method (employing both 80/10/10 and all tokens masking strategies) against the standard random masking baselines (Orig. Hao et al. (2024) and our reproduced $P + FT$). As shown in the tables, MAM consistently outperforms uniform random masking, particularly when using the all-token masking strategy under the decoder or separate adapter context-injection schemes.

Appendix G. Linear Probe Results

Here, we present the full results of the linear probing evaluation across all four datasets. We report the F1-scores for Hematopoietic Niche, PBMC68K, Heart, and Retina in Table 11, Table 12, Table 13, and Table 14, respectively.

Consistent with our main analysis, these results highlight the risks of uniform random masking on complex, sparse data. Specifically, on the Hematopoietic Niche dataset, continued pretraining with random masking causes a severe representation collapse (dropping from 77.36% to 26.51%), whereas MAM avoids this degradation and recovers performance (up to 71.54% with decoder injection). On the smaller or less complex datasets (Heart and Retina), where the baseline performance is already high, MAM remains stable and competitive, demonstrating that our attention-guided masking is a safe and robust strategy for adaptation.

Table 6: Compute resources and end-to-end runtime for every experiment group (MAM). Each experiment used 4 × NVIDIA A40 GPUs (48 GB each).

Dataset	Stage	# Experiments	Approx. Runtime
Hematopoietic niche	Pretrain	10	∼2 weeks
	Finetune (partial)	10	∼1 week
	Finetune (linear probe)	10	∼1 week
	Total	**30**	**∼4 weeks**
PBMC 68K	Pretrain	10	∼1 week
	Finetune (partial)	10	∼0.5 week
	Finetune (linear probe)	10	∼0.5 week
	Total	**30**	**∼2 weeks**
Heart	Pretrain	10	∼0.5 week
	Finetune (partial)	10	∼0.5 week
	Finetune (linear probe)	10	∼0.5 week
	Total	**30**	**∼1.5 weeks**
Retina	Pretrain	10	∼0.5 week
	Finetune (partial)	10	∼0.5 week
	Finetune (linear probe)	10	∼0.5 week
	Total	**30**	**∼1.5 weeks**

Table 7: 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 Hao et al. (2024). 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.Hao et al. (2024) + 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 8: Appendix: Partial-finetune (FT) results on the PBMC68K dataset. Same notation as Table 7. Here Orig. comes from Hao et al. (2024). 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.Hao et al. (2024) + 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 9: Appendix: Partial-finetune (FT) results on the Heart dataset. Same notation as Table 7. Here Orig. comes from Hao et al. (2024). 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.Hao et al. (2024) + 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

Table 10: Appendix: Partial-finetune (FT) results on the Retina dataset. Same notation as Table 7. Here Orig. comes from Hao et al. (2024). 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.Hao et al. (2024) + FT	98.93 ± 0.002
1	Random (80/10/10)	-	P+FT	98.33 ± 0.002
2	Random (All tokens)	-	P+FT	98.47 ± 0.002
3	MAM (80/10/10)	Both	P+FT	99.46 ± 0.002
4	MAM (All tokens)	Both	P+FT	99.44 ± 0.002
5	MAM (80/10/10)	Separate	P+FT	99.08 ± 0.002
6	MAM (All tokens)	Separate	P+FT	98.81 ± 0.002
7	MAM (80/10/10)	Encoder	P+FT	99.20 ± 0.002
8	MAM (All tokens)	Encoder	P+FT	98.44 ± 0.002
9	MAM (80/10/10)	Decoder	P+FT	98.67 ± 0.002
10	MAM (All tokens)	Decoder	P+FT	98.92 ± 0.003

Table 11: Appendix: Linear-probe (LP) results on the Hematopoietic Niche dataset. Same notation as Table 7. Here Orig. comes from Hao et al. (2024). 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.Hao et al. (2024) + 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 12: Appendix: Linear-probe (LP) results on the PBMC68K dataset. Same notation as Table 7. Here Orig. comes from Hao et al. (2024). 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.Hao et al. (2024)+ 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 13: Appendix: Linear-probe (LP) results on the Heart dataset. Same notation as Table 7. Here Orig. comes from Hao et al. (2024). 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.Hao et al. (2024)+ 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

Table 14: Appendix: Linear-probe (LP) results on the Retina dataset. Same notation as Table 7. Here Orig. comes from Hao et al. (2024). 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.Hao et al. (2024)+ LP	84.03 ± 0.003
1	Random (80/10/10)	-	P+LP	97.86 ± 0.004
2	Random (All tokens)	-	P+LP	97.97 ± 0.003
3	MAM (80/10/10)	Both	P+LP	86.37 ± 0.004
4	MAM (All tokens)	Both	P+LP	96.80 ± 0.001
5	MAM (80/10/10)	Separate	P+LP	95.03 ± 0.005
6	MAM (All tokens)	Separate	P+LP	95.15 ± 0.004
7	MAM (80/10/10)	Encoder	P+LP	96.81 ± 0.002
8	MAM (All tokens)	Encoder	P+LP	94.18 ± 0.002
9	MAM (80/10/10)	Decoder	P+LP	94.85 ± 0.003
10	MAM (All tokens)	Decoder	P+LP	95.45 ± 0.004