

additional information of “Unlocking biological insight from single-cell data with an interpretable dual-stream foundation model”

S1. Model Settings and Pre-training Process

The scDMC model consists of 12 Transformer layers, each with 8 attention heads, and a hidden embedding size of 512. The extrapolation size for the ALiBi linear bias attention was set to 2,048, and the vocabulary size was 60,534. During training, a hidden dropout probability of 0.02 and an attention dropout probability of 0.1 were used to prevent overfitting. The model was implemented in PyTorch, utilizing the Hugging Face Transformers library for model configuration, data loading, and training.

The pre-training process for scDMC was conducted on a single A800 GPU, taking approximately 10 days to train for 3 epochs on the 2 million single-cell dataset. We used the AdamW optimizer with a GeLU activation function, an initial learning rate of $5e-5$, and a batch size of 24. Upon completion of training, the model achieved a best loss value of 2.93 on the validation set. The final model parameters were saved to a specified checkpoint path, ready for subsequent fine-tuning or inference on downstream tasks.

The pre-training process shows the evolution of four key loss components of the scDMC model. Fig. S1 displays the training dynamics across all four loss components over the course of optimization. The subplots correspond to the change over training steps for (top left) MLM_rank Loss, (top right) MLM_epe Loss, (bottom left) Contrastive Loss, and (bottom right) Total Loss. All loss components exhibit a healthy downward trend, indicating that the model is learning effectively and converging. The MLM_epe Loss is much smaller in magnitude than the other losses; to better observe its dynamics, its Y-axis is shown on a logarithmic scale. The steady decline of all metrics confirms the stability and effectiveness of the pre-training strategy employed by the scDMC model.

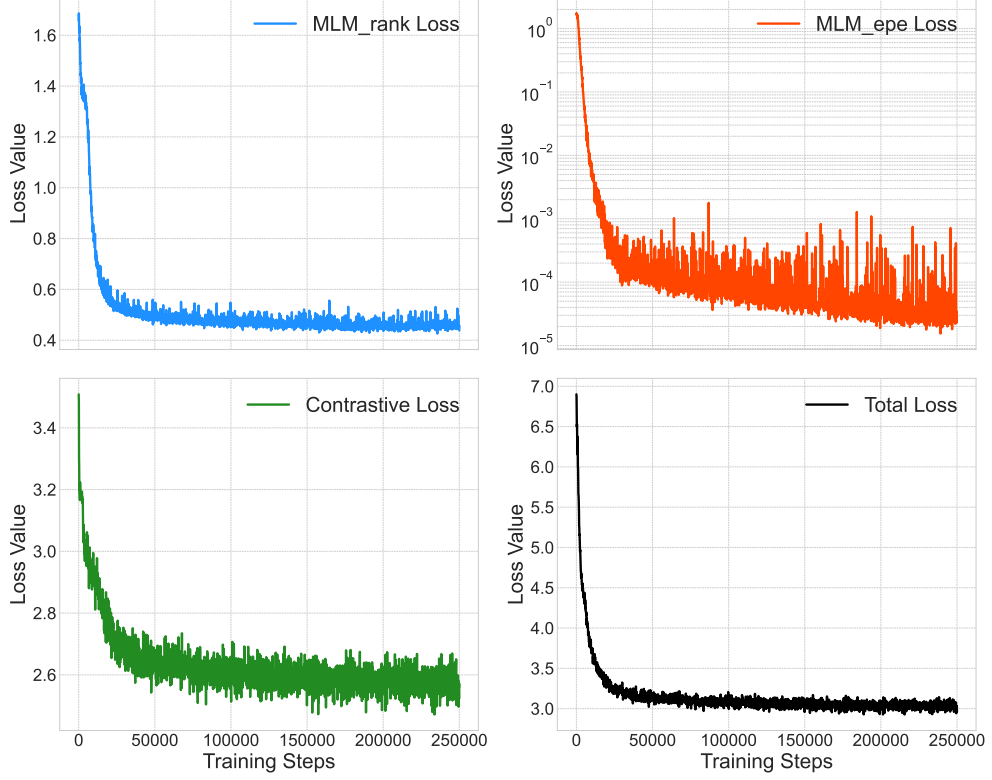


Fig. S1 Analysis of loss components during the scDMC pre-training process.

S2. Detailed workflow for gene module discovery and functional enrichment analysis

To identify functionally related gene modules, we employed the following workflow, which was applied to two representative cell types:

(1) Gene Embedding Extraction. First, from an scDMC model fine-tuned on a specific dataset (e.g., Zheng68K), we extracted the average contextual gene embeddings for all expressed genes (12,535 in total) within a specific cell type (e.g., CD8⁺ cytotoxic T cells). These embedding vectors were organized into a feature matrix with genes as rows and embedding dimensions as columns.

(2) Gene Module Clustering. Subsequently, we applied the unsupervised graph-based clustering algorithm Leiden (from the leidenalg Python package, with the resolution parameter set to 0.2) to the aforementioned gene embedding matrix. This was done to identify functionally related groups of genes, or “gene modules”. In CD8⁺ T cells, this procedure identified 8 major gene modules (Main Fig. 5a).

(3) Functional Enrichment Analysis. We used the online gene enrichment analysis tool Metascape (<https://metascape.org/>) to functionally annotate each identified gene module. For the largest module identified in CD8⁺ T cells (Module 0,

containing 3,836 genes), its size exceeded Metascape’s input limit of 3,000 genes. We inferred that this module likely represents housekeeping genes involved in basic cellular activities with dispersed functions. Therefore, it was excluded from functional enrichment analysis, and we focused on the remaining 7 more functionally specific modules (Main Fig. 5c).

(4) Cross-Dataset Validation. To validate the robustness of this workflow, we repeated steps 1 through 3 verbatim on hematopoietic stem cells (HSCs) from the BMDC dataset. The results similarly identified gene modules with high functional cohesion (Main Fig. 5d), demonstrating the generalizability of our analytical approach.

S3. Gene Regulatory Network (GRN) Construction and Hub Gene Identification

A central challenge in constructing a GRN is determining an appropriate similarity threshold. To address this, we systematically evaluated how key network topological properties evolve as a function of the cosine similarity threshold, ranging from 0.5 to 0.9 (Fig. S2). The objective of this analysis is to identify an optimal threshold that effectively filters out weak connections (likely noise) while preserving the core network of strong, biologically meaningful interactions.

In two distinct cell types and datasets, we consistently observed a clear “phase transition” region with a critical point near the similarity threshold of 0.7 (marked by a red dashed line)(Fig. S2). At this critical point, the number of edges (top row) and the average clustering coefficient (middle row)both decrease sharply, indicating the removal of a large number of weak-to-moderate connections. Crucially, while the size of the largest connected component (bottom row) also contracts sharply, it stabilizes into a substantial core network rather than completely disintegrating. This phenomenon indicates that the 0.7 threshold acts as an inflection point for separating strong biological signals from noise. Based on this robust, cross-dataset observation, we therefore consistently applied 0.7 as the similarity threshold for all GRN analyses in this study. Hub genes within these networks were subsequently identified by calculating their degree centrality (Table S1).

Table S1: Top 10 hub genes in CD8⁺ T cells by degree centrality.

Rank	Gene Symbol	Degree Centrality
1	<i>SPX3</i>	0.0537
2	<i>FAM153A</i>	0.0491
3	<i>C9ORF34</i>	0.0375
4	<i>TFIIS.H</i>	0.0373
5	<i>TBS2</i>	0.0373
6	<i>MCE</i>	0.0350
7	<i>DISPA</i>	0.0348
8	<i>IP6K</i>	0.0331
9	<i>TAM41</i>	0.0280
10	<i>ZNF439</i>	0.0276

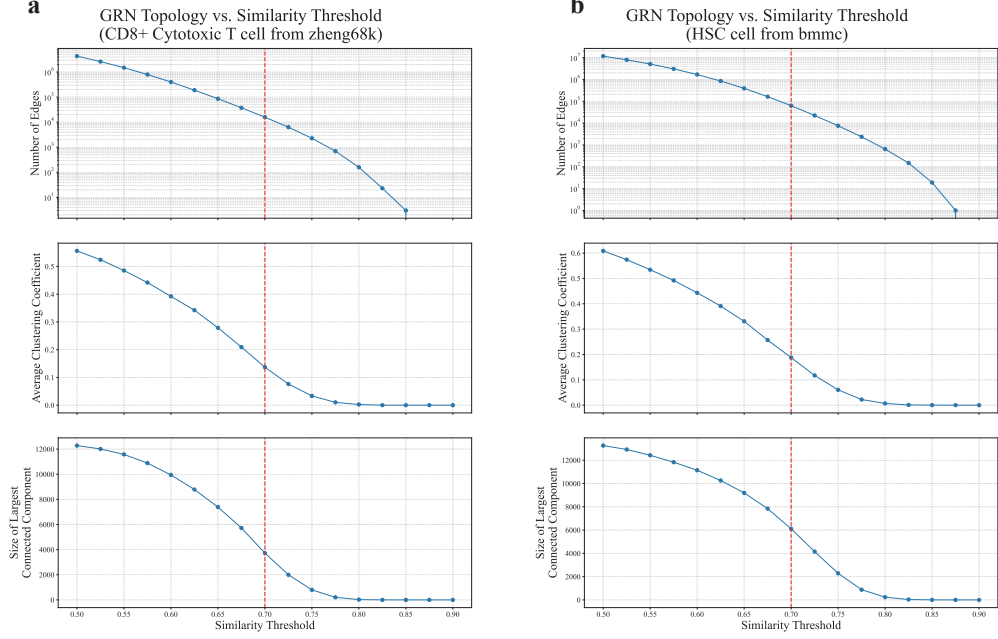


Fig. S2 Selecting the similarity threshold for Gene Regulatory Networks (GRNs) based on network topology. **a**, Analysis performed on the GRN from CD8⁺ cytotoxic T cells in the Zheng68k dataset. **b**, The same analysis performed on the GRN from hematopoietic stem cells (HSCs) in the BMME dataset. In each panel, the plots from top to bottom show: **Number of Edges**, plotted on a logarithmic scale, reflecting network density; **Average Clustering Coefficient**, a measure of network modularity (the tendency of nodes to form tightly connected groups); **Size of Largest Connected Component**, reflecting overall network connectivity.

S4. Discovering and Validating Functional Gene Programs from Fine-Tuned Gene Embeddings

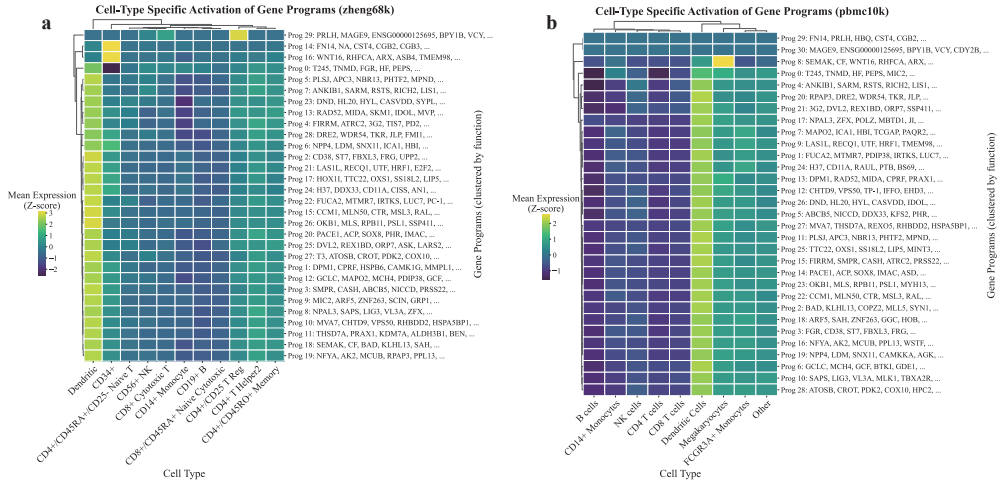
To validate our core hypothesis that the fine-tuning process refines the model’s understanding of gene semantics, we designed an analysis to discover functionally related “gene programs” from fine-tuned gene embeddings in a data-driven manner. Specifically, from an scDMC model that was fine-tuned for a particular downstream task (e.g., cell type annotation), we extracted the contextual embeddings for all genes. Subsequently, we applied the unsupervised clustering algorithm Leiden to group these gene embeddings. To evaluate the biological significance of these data-driven gene programs, we calculated the average activation score (Z-score) for each program across various cell types and visualized the results as a heatmap (Fig. S3).

The analysis revealed highly specific associations between gene programs and cell types, which are in strong agreement with established biological knowledge. In the

Zheng68k dataset (Fig. S3a), the synergistic, high activation of numerous gene programs in Dendritic Cells perfectly reflects their “multi-threaded” working mode as “professional” antigen-presenting cells. Furthermore, the specific activation of Program 16, which includes the *WNT16* gene, in CD8⁺ cells demonstrates the model’s ability to capture core regulatory pathways essential for maintaining stem cell identity.

This finding was validated in the independent PBMC10k dataset (Fig. S3b). A particularly striking example is the strong activation of nearly all gene programs in Megakaryocytes, which accurately portrays their global transcriptional state, geared for massive biosynthesis as “platelet factories”.

These analyses demonstrate that clustering the fine-tuned gene embeddings from scDMC can reliably identify gene programs that reflect the core identity and functional state of cells, providing a powerful tool for interpretable biological discovery using deep learning models.



S5. Datasets

S5.1 Pre-training Dataset

The pre-training dataset was sourced from the CELLxGENE Census database and comprises 2 million healthy human blood cells. We obtained the data from the CELLxGENE portal (<https://cellxgene.cziscience.com/>) using the Census API (<https://chanzuckerberg.github.io/cellxgene-census/python-api.html>) to query the “blood” tissue from the CELLxGENE Census (release version 2024-07-01), which yielded over 5.4 million single-cell blood records. To demonstrate the efficiency of the scDMC framework under limited computational resources, we selected the first 2 million cells from this collection and constructed the pre-training dataset using our tokenizer and vocabulary. During tokenization, we standardized only the gene expression matrix, without performing any additional cell quality control or dimensionality reduction steps. The pre-training dataset includes 131 cell types from 6,603 healthy human donors, of whom 67.3% were male.

The distribution of major cell types is shown in the treemap in Fig. S4, where the primary populations are “CD14-positive monocyte”, “natural killer cell”, and “central memory CD4-positive, alpha-beta T cell”. These three cell types constitute the most abundant and dominant populations in the dataset. Secondary populations, such as “B cell” and “monocyte”, occupy moderately sized rectangles, indicating that they also represent a substantial proportion of the sample. Cell heterogeneity is further visualized via UMAP in Fig. S5.

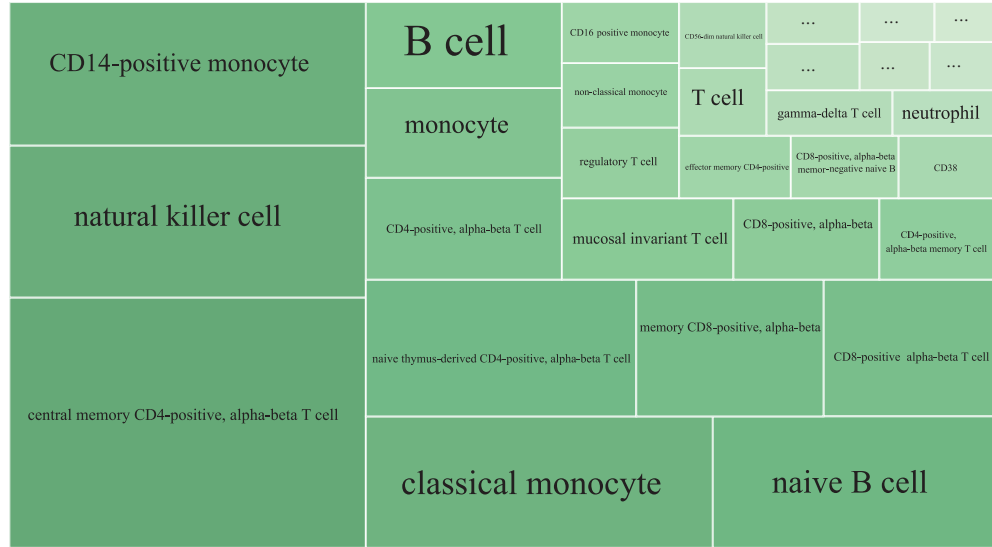


Fig. S4 Distribution of major cell types in the pre-training dataset. Each rectangle in the treemap represents a specific cell type, with its area proportional to the relative abundance of that cell type.

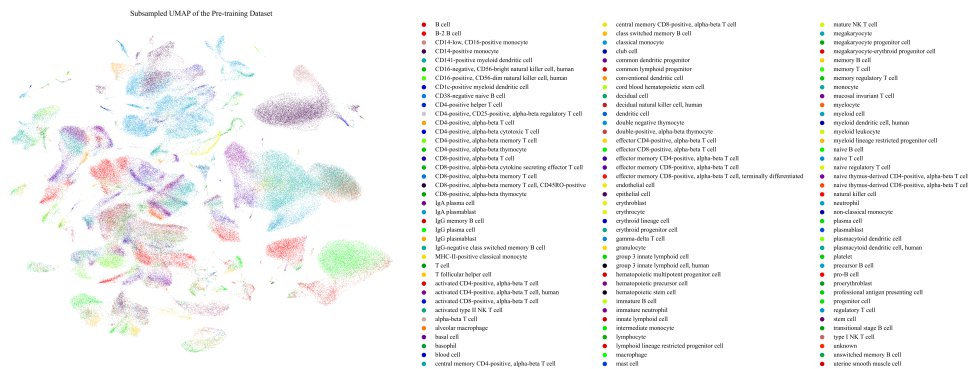


Fig. S5 Visualization of cell heterogeneity in the pre-training dataset. The UMAP plot shows a 10% random subsample of cells, with different colors representing different cell type identities.

S5.2 Downstream Task Datasets

- **Immune Human:** A comprehensive immune cell atlas containing 33,506 cells and integrating 5 scRNA-seq datasets from bone marrow and peripheral blood. This dataset encompasses multiple sequencing technologies (10x v2/v3, Smart-seq2), 16 cell types, and 10 technology batches, making it an ideal benchmark for evaluating model performance in cell type annotation and cross-platform batch integration.
- **Zheng68K:** A classic dataset of approximately 68,000 human peripheral blood mononuclear cells (PBMCs). It is characterized by its imbalanced cell type distribution and high similarity between certain subtypes, including several rare cell types (e.g., CD8⁺ progenitor cells), posing a stringent challenge for rare-cell identification and fine-grained classification.
- **PBMC10k:** A multi-batch dataset of approximately 12,000 PBMCs from two batches. It is commonly used to evaluate a model’s ability to integrate batch effects. We use it to test whether scDMC can maintain a clear distinction among 9 immune cell subtypes while removing batch differences.
- **Bone Marrow Mononuclear Cells (BMMC, CITE-seq):** A multimodal dataset of over 90,000 human bone marrow mononuclear cells. We use only its gene expression (GEX) component. It is sourced from 12 healthy donors and contains 45 finely annotated immune cell subtypes, providing a rigorous benchmark for multi-donor batch integration and ultra-high-resolution cell annotation.
- **MS (Multiple Sclerosis):** A single-cell dataset from brain tissue of multiple sclerosis patients and healthy controls. This dataset includes both healthy and diseased states and samples from different brain regions (e.g., prefrontal cortex), making it suitable for testing model robustness and representation capabilities on disease-related data.
- **Myeloid (Pan-cancer):** A pan-cancer myeloid cell dataset containing samples from tumor and adjacent normal tissues across 9 different cancer types. Following the setup of the original study, we divided it into a reference set (6 cancer types) and

a query set (3 cancer types) to evaluate the model’s generalization and annotation capabilities across different cancer types and tissue sources.

- **Pancreas (Human Pancreas Atlas):** A comprehensive single-cell atlas of the human pancreas containing 16,382 cells, integrating multiple classic scRNA-seq datasets across 9 technology batches (e.g., Smart-seq2, CEL-seq, inDrop). This dataset encompasses 14 distinct pancreatic cell types (including Alpha, Beta, Delta, Gamma, and Ductal cells) from both healthy donors and patients with type 2 diabetes. As a complex, multi-batch dataset from a non-hematopoietic endocrine organ, it serves as a rigorous benchmark for evaluating cross-organ generalization and transfer learning beyond the blood system.

Table S2: Detailed information for downstream task datasets.

Dataset	Tissue	Disease	Batches	Cell Types	Cells
Immune	Marrow and Blood	Normal	10	16	33,506
Zheng68K	Blood	Normal	1	11	68,263
PBMC10K	Blood	Normal	2	9	11,989
BMMC	Marrow	Normal	12	45	90,261
MS	Brain	MS	2	18	21,312
Myeloid	Marrow	Pan-cancer	2	21	13,178
Pancreas	Pancreas	Normal	9	14	16,382

S6. Evaluation Metrics

To comprehensively evaluate the performance of our model, we employed a variety of established evaluation metrics tailored to different downstream tasks. These metrics are primarily used to quantify and assess the model’s performance in supervised cell type annotation, unsupervised cell clustering, and batch integration.

S6.1 Cell Type Annotation Metrics

Cell type annotation is a supervised classification task where the goal is to assign a predefined type label to each cell. We use standard classification metrics: Accuracy, Precision, Recall, and Macro F1-Score to evaluate the cell type assignment performance. Assuming the set of true classes in a dataset is C , for any class $c \in C$, we define True Positives (TP_c), False Positives (FP_c), and False Negatives (FN_c).

- **Accuracy:** Accuracy is the most intuitive performance metric, measuring the proportion of all cell samples that are correctly classified by the model.

$$\text{Accuracy}_c = \frac{TP_c + TN_c}{TP_c + TN_c + FP_c + FN_c} \quad (1)$$

- **Precision:** For a class c , precision measures how many of the samples predicted as class c truly belong to that class.

$$\text{Precision}_c = \frac{TP_c}{TP_c + FP_c} \quad (2)$$

- **Recall:** For a class c , recall measures how many of all true samples belonging to that class were correctly predicted.

$$\text{Recall}_c = \frac{TP_c}{TP_c + FN_c} \quad (3)$$

- **F1-Score:** The F1-Score is the harmonic mean of precision and recall, providing a combined measure of their performance.

$$F1_c = 2 \cdot \frac{\text{Precision}_c \cdot \text{Recall}_c}{\text{Precision}_c + \text{Recall}_c} \quad (4)$$

- **Macro F1-Score:** To evaluate the model’s performance equally across all classes, especially in datasets with class imbalance, we calculate the Macro F1-Score. It is the arithmetic mean of the F1-Scores for all classes.

$$\text{Macro F1} = \frac{1}{|C|} \sum_{c \in C} F1_c \quad (5)$$

S6.2 Unsupervised Clustering and Batch Integration Metrics

For unsupervised clustering and batch integration tasks, we evaluate the quality of the cell embeddings generated by the model. These metrics are largely based on the scIB benchmark framework and assess performance from two complementary dimensions: conservation of biological structure and correction of batch effects.

First are three metrics for the conservation of biological structure, which measure the extent to which the cell embeddings retain meaningful biological differences, such as distinguishing between different cell types.

To ensure a rigorously fair evaluation, we implemented an Adaptive Resolution Sweep strategy. Instead of relying on a default parameter, we performed a comprehensive grid search over the resolution parameter space ranging from 0.1 to 2.0 (with a step size of 0.1) independently for each model on every dataset. For each candidate resolution, we computed the Adjusted Rand Index (ARI) and Normalized Mutual Information (NMI) against the ground-truth cell type labels. We then identified the "optimal resolution" for each model that maximized the combined metric: $(\text{ARI} + \text{NMI})/2$.

By reporting the metrics obtained at this optimal resolution, we effectively compare the theoretical "upper bound" or peak clustering performance of each embedding space. This methodology completely isolates the intrinsic discriminative quality of the learned representations from the confounding effects of suboptimal hyperparameter selection.

- **Adjusted Rand Index (ARI):** ARI measures the similarity between two data clusterings. Here, we compare the algorithm-generated cluster labels K with the true cell type labels C . The range of ARI is $[-1, 1]$, where 1 indicates a perfect match and 0 indicates the result is no better than random assignment.

$$\text{ARI} = \frac{\text{RI} - E[\text{RI}]}{\max(\text{RI}) - E[\text{RI}]} \quad (6)$$

where RI is the Rand Index and $E[\text{RI}]$ is its expected value.

- **Normalized Mutual Information (NMI):** NMI measures the mutual dependence between two label assignments, independent of the actual label values. Its range is, where 1 indicates perfect agreement between the two label assignments. Let the true labels be Y and the clustering results be C .

$$\text{NMI}(Y, C) = \frac{2 \cdot I(Y; C)}{H(Y) + H(C)} \quad (7)$$

where $I(Y; C)$ is the mutual information, and $H(Y)$ and $H(C)$ are the entropies of the label distributions.

- **Cell Type Average Silhouette Width (ASW):** The silhouette width measures how tightly grouped a sample is within its own cluster compared to other clusters. For a single sample, its silhouette coefficient s is defined as:

$$s = \frac{b - a}{\max(a, b)} \quad (8)$$

where a is the mean distance from the sample to all other points in the same cluster, and b is the mean distance from the sample to all points in the next nearest cluster. We average the silhouette coefficients (based on true cell type labels) for all cells to get $\text{ASW}_{\text{cell-type}}$. To normalize its range to, we perform the following transformation:

$$\text{ASW}_{\text{cell}} = \frac{\text{ASW}_{\text{cell-type}} + 1}{2} \quad (9)$$

A higher ASW_{cell} score indicates better separation of cell types in the embedding space.

Next are several evaluation metrics for batch effect correction, specifically designed to assess batch integration tasks by measuring the extent to which the model removes technical batch differences.

- **Batch Average Silhouette Width (ASW):** Similar to the cell type ASW, this metric calculates the average silhouette width based on batch labels, $\text{ASW}_{\text{batch-label}}$. However, its ideal value is -1 (indicating perfect mixing of cells from different batches). We transform it into a metric with a range of, where a higher score represents better batch mixing.

$$\text{ASW}_{\text{batch}} = 1 - |\text{ASW}_{\text{batch-label}}| \quad (10)$$

- **Graph Connectivity:** This metric assesses whether cells of each cell type maintain their connectivity in a k-nearest neighbor (k-NN) graph. For each cell type c , we calculate the size of its largest connected component (LCC) and divide it by the total number of cells of that type, N_c . The final score is the average over all cell types.

$$\text{GraphConn} = \frac{1}{|C|} \sum_{c \in C} \frac{|\text{LCC}(G_c^{kNN})|}{N_c} \quad (11)$$

A higher score indicates that the local neighborhood structure of cells is better preserved after integration.

To obtain an overall performance assessment, we aggregate the above metrics to produce a final composite score:

$$\text{Overall Score} = 0.6 \cdot \text{AVG}_{\text{Bio}} + 0.4 \cdot \text{AVG}_{\text{Batch}} \quad (12)$$

where AvgBio is the average of the biological conservation metrics (ARI, NMI, and ASW_{cell}), and $\text{AVG}_{\text{Batch}}$ is the average of the batch correction metrics ($\text{ASW}_{\text{batch}}$, GraphConn). The Overall Score balances the two objectives of conserving biological structure and removing batch effects.

S7. Additional Experimental Results

S7.1 Additional Results for Cell Type Annotation

Table S3 shows the performance comparison of various models on the cell type annotation task. On the Zheng68k dataset, scDMC’s performance highlights how its dual-stream pre-training framework learns the features of major cell populations while enhancing the discrimination of low-frequency signals (i.e., rare cell types) through contrastive learning. This is evidenced in the confusion matrix comparison in Fig. S6, where the matrices are row-normalized to show per-class recall. For rare cell types such as CD8^+ cells, scDMC achieved not only a recall of 0.91 but also a high precision of 0.93, significantly outperforming other models and signifying the higher reliability of its predictions. Furthermore, when handling easily confusable classes (e.g., $\text{CD4}^+/\text{CD45RO}^+$ Memory cells), scDMC also demonstrated the most balanced performance, avoiding the significant performance shortcomings seen in other models for this class. This comprehensive advantage ultimately overcomes the bias that models tend to develop on imbalanced data, ensuring the precise identification of all cell sub-populations. In clinical or biological research, rare cell types (such as cancer stem cells or specific immunoregulatory cells) often play a critical role. Therefore, this capability of scDMC is not merely a technical improvement but also signifies its immense potential for application in translational medicine research.

In our standard scDMC downstream fine-tuning protocol, the model relies on the rank-based stream ($\mathbf{S}_{\text{rank}}$), where non-zero expressed genes are arranged in descending order based on their normalized expression levels. This specific ordering is highly informative, as it provides the model with rich biological semantics regarding the relative importance of each gene within a specific cellular state. Furthermore, this

Table S3: Performance comparison of various models on the cell type annotation task.

Models	Zheng68K		PBMC10K		BMMC		MS		Pancreas	
	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1
scBert	0.7982	0.6952	0.9554	0.8613	0.8240	0.5425	0.8365	0.7316	0.9357	0.6822
scGPT	0.8077	0.6706	0.9575	0.8615	0.8301	0.5699	0.8329	0.6356	0.9432	0.7024
Geneformer	0.8215	0.7034	0.9604	0.8228	0.8202	0.5128	0.7703	0.6645	0.9710	0.7749
Geneformer-95M	0.8456	0.7378	0.9733	0.9432	0.8543	0.6075	0.8787	0.7898	0.9762	0.8898
scFoundation	0.7316	0.6297	0.9716	0.9422	0.8375	0.5925	0.8278	0.8057	0.9792	0.8913
LangCell	0.8552	0.7459	0.9722	0.9502	0.8540	0.6171	0.8705	0.8004	0.9811	0.8708
scDMC	0.8660	0.7674	0.9708	0.9522	0.8511	0.6689	0.8792	0.8278	0.9811	0.9404

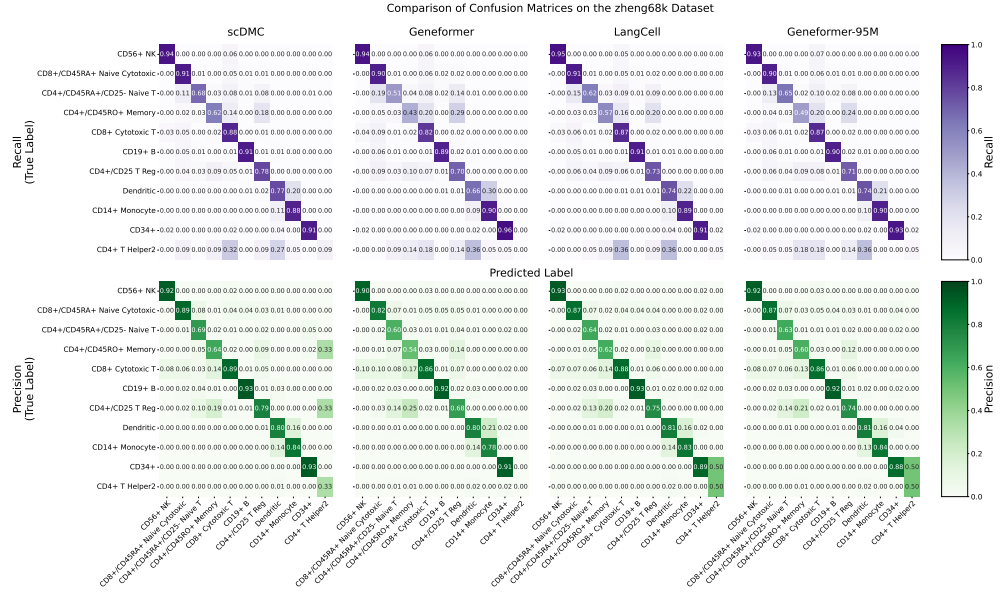


Fig. S6 Performance of various models on the Zheng68k dataset, shown by row-normalized confusion matrices.

ranking provides a meaningful structural prior for the Attention with Linear Biases (ALiBi) mechanism to compute relative distances.

To explicitly isolate and evaluate the impact of this ordering, we designed a “Random Gene Order” ablation variant for the fine-tuning phase. In this variant, instead of ranking genes by their expression levels, we completely randomized the sequence order of the non-zero expressed genes for each individual cell before feeding them into the pre-trained scDMC model. We then evaluated this randomized variant on the cell type annotation task using the PBMC10k and the highly challenging Zheng68K datasets.

As shown in Table S4, randomly shuffling the gene sequence during fine-tuning resulted in a noticeable degradation in downstream classification performance across both datasets. This result firmly confirms the crucial role of meaningful sequence ordering for the Transformer architecture to effectively extract discriminative cellular features. Specifically, this ordering conveys the relative biological importance of genes and provides a stable structural prior for the ALiBi mechanism. However, it is also worth noting that despite the complete disruption of sequence order, the model maintained a robust baseline performance, which underscores the strong generalized feature extraction capabilities embedded within the pre-trained scDMC weights.

Table S4: Performance impact of randomizing input gene order during scDMC fine-tuning. The random variant was evaluated across 3 independent runs with different random seeds. Results are presented as Mean $\pm$ Standard Deviation (Std).

Dataset	Input Variant	Accuracy	Macro F1-Score
PBMC10k	Random	0.9437 $\pm$ 0.0066	0.7778 $\pm$ 0.0123
	Ranked	0.9708	0.9522
Zheng68K	Random	0.8580 $\pm$ 0.0170	0.7553 $\pm$ 0.0130
	Ranked	0.8660	0.7674

S7.2 Additional Results for Batch Integration

According to the experimental results in Table S5, the scDMC model demonstrates comprehensive and significant superiority in cell clustering and batch integration tasks across several other single-cell datasets (BMMC, Immune, PBMC10K, MS, Myeloid). The model not only far surpasses the LangCell and Geneformer models on key clustering metrics that measure the conservation of biological state, such as Adjusted Rand Index (ARI), Normalized Mutual Information (NMI), and cell type Average Silhouette Width (ASW_{cell}) achieving the most precise partitioning of cell subpopulations, but it also performs best on metrics that evaluate the removal of technical effects, such as the batch Average Silhouette Width (ASW_{batch}) and Graph Connectivity (GraphConn), effectively eliminating inter-batch differences. Ultimately, its Overall Score ranked first in all tests, fully demonstrating that scDMC can maximally preserve true biological heterogeneity while efficiently integrating multi-batch data.

S7.3 Computational Cost and Efficiency Analysis

To evaluate the practical utility and computational overhead of our framework, we benchmarked the downstream efficiency of scDMC against representative single-cell foundation models, including scGPT and scFoundation.

It is crucial to clarify that the momentum contrastive learning mechanism, encompassing the momentum encoder and the expression-aware stream (S_{epe}), is deployed

exclusively during the pre-training phase. During downstream fine-tuning and inference (e.g., cell type annotation), scDMC operates as a standard, efficient encoder-only architecture. It takes only the rank-based sequence ($\mathbf{S}_{\text{rank}}$) as input and optimizes standard task-specific losses (e.g., cross-entropy), without any positive/negative sample construction or contrastive loss computation. Consequently, the contrastive learning design introduces no additional computational burden during downstream inference. Furthermore, during pre-training, the momentum encoder’s parameters are updated via an exponential moving average (EMA). Because this EMA update requires no gradient computation or backpropagation, the additional pre-training overhead is tightly controlled (requiring approximately 1.5 times the computational cost of a standard single-stream forward pass, rather than a full 2-fold increase).

Table S5: Additional results for cell clustering and batch integration.

Dataset	Model	Leiden (res)	ARI	NMI	ASW_cell	AvgBio	ASW_batch	GraphConn	Avg_batch	Overall
BMMC	Geneformer	0.4	0.5914	0.7957	0.6632	0.6834	0.8774	0.9141	0.8957	0.7683
	LangCell	0.2	0.6037	0.8010	0.6812	0.6953	0.8886	0.9329	0.9107	0.7815
	scFoundation	0.1	0.6391	0.7499	0.5624	0.6505	0.8867	0.9238	0.9052	0.7524
	scGPT	0.3	0.4072	0.7031	0.5451	0.5518	0.8418	0.7827	0.8122	0.6560
	scDMC	0.5	0.9668	0.9355	0.7959	0.8994	0.8994	0.9590	0.9292	0.9113
Immune	Geneformer	0.1	0.8546	0.8234	0.7473	0.8084	0.8602	0.9638	0.9120	0.8498
	LangCell	0.1	0.6378	0.7893	0.7612	0.7295	0.8695	0.9826	0.9261	0.8081
	scFoundation	0.3	0.4029	0.6562	0.5729	0.5440	0.8233	0.8983	0.8608	0.6707
	scGPT	0.1	0.6274	0.7690	0.6001	0.6657	0.8005	0.9703	0.8853	0.7536
	scDMC	0.2	0.9331	0.9046	0.8201	0.8859	0.8700	0.9890	0.9295	0.9034
PBMC10K	Geneformer	0.3	0.9158	0.8541	0.8310	0.8670	0.9603	0.9440	0.9522	0.9010
	LangCell	0.1	0.8281	0.8676	0.8370	0.8442	0.9564	0.9848	0.9706	0.8948
	scFoundation	0.1	0.8618	0.8306	0.6120	0.7681	0.9704	0.9291	0.9498	0.8408
	scGPT	0.2	0.6919	0.8038	0.7397	0.7451	0.9499	0.9253	0.9376	0.8221
	scDMC	0.1	0.9835	0.9689	0.9096	0.9540	0.9533	0.9907	0.9720	0.9612
MS	Geneformer	0.3	0.5950	0.7138	0.6285	0.6458	0.9229	0.9315	0.9272	0.7584
	LangCell	0.5	0.8272	0.8554	0.6997	0.7941	0.9258	0.9637	0.9448	0.8544
	scFoundation	0.2	0.2780	0.5355	0.5100	0.4412	0.9183	0.9105	0.9144	0.6304
	scGPT	0.3	0.4510	0.6806	0.5848	0.5721	0.9103	0.9218	0.9161	0.7097
	scDMC	0.8	0.9447	0.9438	0.8404	0.9096	0.9422	0.9771	0.9596	0.9296
Myeloid	Geneformer	0.6	0.5116	0.5758	0.5576	0.5483	0.8799	0.8447	0.8623	0.6739
	LangCell	0.3	0.5704	0.6323	0.5780	0.5936	0.8954	0.8578	0.8766	0.7068
	scFoundation	0.7	0.3558	0.5049	0.5032	0.4546	0.8653	0.8996	0.8824	0.6257
	scGPT	0.4	0.3507	0.5068	0.5353	0.4642	0.8506	0.8950	0.8726	0.6276
	scDMC	0.4	0.9038	0.8773	0.7769	0.8527	0.9113	0.9415	0.9264	0.8822
Pancreas	Geneformer	0.1	0.9166	0.8954	0.8453	0.8857	0.8328	0.9009	0.8668	0.8782
	LangCell	0.3	0.9827	0.9624	0.8768	0.9406	0.8622	0.9543	0.9082	0.9277
	scFoundation	0.4	0.2760	0.4574	0.5031	0.4122	0.8153	0.8696	0.8425	0.5843
	scGPT	0.1	0.6046	0.7790	0.7316	0.7051	0.8512	0.9278	0.8895	0.7789
	scDMC	0.3	0.9938	0.9859	0.9503	0.9766	0.8325	0.9907	0.9116	0.9506

To empirically demonstrate its downstream efficiency, we evaluated the fine-tuning and inference speed of scDMC, scGPT, and scFoundation on the Zheng68K dataset (approx. 68,000 cells) using a single NVIDIA RTX 5090 GPU (32GB VRAM). Crucially, to ensure a rigorously fair ”apples-to-apples” comparison of intrinsic architectural efficiency, we uniformly truncated the maximum input sequence length for token-based models (scDMC and scGPT) to 1,024 tokens, and the batch size was identically set to 8 across all models.

As detailed in Table S6, scDMC achieves the fastest fine-tuning throughput (414.70 s/epoch), significantly outperforming both scFoundation and scGPT. In terms of inference speed, scDMC (27.55 s/10k cells) is nearly three times faster than scFoundation and maintains a highly competitive efficiency on par with scGPT.

It is worth noting the engineering trade-offs observed in these benchmarks. While scFoundation exhibits the lowest Peak VRAM (2.64 GB), our code-level analysis reveals that its official fine-tuning script freezes the majority of its parameters (the first 10 Performer layers), drastically reducing optimizer state memory. In contrast, scDMC undergoes full-parameter fine-tuning. Furthermore, scFoundation’s dynamic sequence construction for all non-zero genes introduces substantial computational overhead, neutralizing the theoretical speed advantage of its $O(N)$ Performer attention. In contrast, scDMC leverages a streamlined Encoder-only architecture combined with FlashAttention and a fixed sequence truncation. These empirical results conclusively demonstrate that the robust representations learned via our dual-stream contrastive pre-training seamlessly translate to highly efficient, full-parameter downstream deployment.

Table S6: Comparison of downstream computational cost and efficiency across single-cell foundation models. (Evaluated on the Zheng68K dataset with a batch size of 8. Fine-tuning time is measured per epoch, and inference time is measured per 10,000 cells.)

Model	Architecture	Vocab Size	Parameters	Peak VRAM	Fine-tuning Time	Inference Time
scFoundation	xTrimoGene (Performer) Enc-Dec	19264	100.14M	2.64GB	636.45 s/e-poch	74.94 s/10k cells
scGPT	GPT-style (Decoder-only)	60697	51.34M	3.03GB	499.72 s/e-poch	18.72 s/10k cells
scDMC	BERT-style (Encoder-only)	60534	57.55M	3.48GB	414.70 s/e-poch	27.55 s/10k cells

Note 1 (scFoundation): Low Peak VRAM due to freezing first 10 encoder layers during fine-tuning, while scDMC and scGPT use full-parameter fine-tuning.

Note 2 (scGPT): Parameters correspond to scGPT_human (Jun 23, 2023) with vocab 60,697 (CELLxGENE Census), ~51.3M params (vs. original ~33M), ensuring fair comparison with scDMC.

S7.4. Comparison of End-to-End Contrastive Pre-training versus Post-hoc Embedding Fusion

To determine whether the performance gains of scDMC could be achieved by simply aligning or combining the outputs of existing models utilizing different tokenization schemes (as hypothesized by reviewers), we conducted a Post-hoc Fusion (late fusion) experiment on the Immune human dataset.

From a representation learning perspective, post-hoc fusion operates on frozen embedding vectors that have been independently processed by single-stream encoders. In such separated architectures, the self-attention heads cannot share synergistic information across different modalities during deep feature extraction. In contrast,

scDMC’s end-to-end momentum contrastive learning dynamically forces the underlying attention mechanisms to identify and align cross-view features during the backpropagation process of pre-training.

To empirically validate this, we extracted the [CLS] embeddings from our independently trained RankOnly and EPEOnly models for each cell in the Immune dataset. We concatenated these two 512-dimensional vectors to form a 1024-dimensional post-hoc fused feature vector. Subsequently, we performed a rigorous Linear Probing evaluation by training an L2-regularized multinomial Logistic Regression classifier (solved to full convergence) on these frozen fused embeddings.

As detailed in Table S7, the Post-hoc Fusion strategy yields a Macro F1-score of 0.8631. While this marginally improves upon the individual single-stream variants in terms of Macro F1 (confirming that the two tokenizers indeed offer complementary information), its overall accuracy actually decreases compared to the EPEOnly model, indicating the introduction of unaligned feature redundancy. Crucially, its performance remains significantly inferior to the complete, end-to-end optimized scDMC model (Accuracy: 0.9364, Macro F1: 0.9167). This striking contrast unequivocally answers the architectural question: simply aligning or fusing embeddings from existing independent models is highly suboptimal. It lacks the deep, cross-view attention synergy that our dual-stream architecture captures during end-to-end backpropagation. This underscores the irreplaceable value and unique contribution of scDMC’s contrastive framework in shaping a highly discriminative representational manifold.

Table S7: Performance comparison of end-to-end contrastive pre-training versus post-hoc embedding fusion on the Immune human dataset.

Model Strategy	Tokenization Scheme	Accuracy	Macro F1
RankOnly	Rank-based only (End-to-end fine-tuning)	0.9091	0.7566
EPEOnly	Expression-aware only (End-to-end fine-tuning)	0.9236	0.8573
Post-hoc Fusion	Rank + EPE (Frozen feature concatenation)	0.9049	0.8631
RankOnly	Dual-stream with momentum contrastive alignment	0.9364	0.9167