

# scMamba: A Pre-Trained Model for Single-Nucleus RNA Sequencing Analysis in Neurodegenerative Disorders

Gyutaek Oh<sup>1,\*</sup>, Baekgyu Choi<sup>2,\*</sup>, Seyoung Jin<sup>2</sup>, Inkyung Jung<sup>2,†</sup>, and Jong Chul Ye<sup>3,†</sup>

<sup>1</sup>*Department of Bio and Brain Engineering, Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Republic of Korea*

<sup>2</sup>*Department of Biological Sciences, Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Republic of Korea*

<sup>3</sup>*Kim Jaechul Graduate School of AI, Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Republic of Korea*

<sup>†</sup>*Correspondence should be addressed to Jong Chul Ye (jong.ye@kaist.ac.kr) or Inkyung Jung (ijung@kaist.ac.kr).*

## Abstract

Single-nucleus RNA sequencing (snRNA-seq) has significantly advanced our understanding of the disease etiology of neurodegenerative disorders. However, the low quality of specimens derived from postmortem brain tissues, combined with the high variability caused by disease heterogeneity, makes it challenging to integrate snRNA-seq data from multiple sources for precise analyses. To address these challenges, we present scMamba, a pre-trained model designed to improve the quality and utility of snRNA-seq analysis, with a particular focus on neurodegenerative diseases. Inspired by the recent Mamba model, scMamba introduces a novel architecture that incorporates a linear adapter layer, gene embeddings, and bidirectional Mamba blocks, enabling efficient processing of snRNA-seq data while preserving information from the raw input. Notably, scMamba learns generalizable features of cells and genes through pre-training on snRNA-seq data, without relying on dimension reduction or selection of highly variable genes. We demonstrate that scMamba outperforms benchmark methods in various downstream tasks, including cell type annotation, doublet detection, imputation, and the identification of differentially expressed genes.

## Introduction

Single-cell RNA sequencing (scRNA-seq) is a powerful technique for profiling gene expression at single-cell resolution, enabling the exploration of molecular characteristics in complex biological systems under both normal and disease conditions <sup>1,2</sup>. scRNA-seq analysis facilitates several key objectives, including cell type annotation <sup>3,4</sup>, the discovery of novel cell types <sup>5</sup>, the identification of marker genes <sup>6</sup>, and the analysis of cellular heterogeneity <sup>7,8</sup>.

Notably, the brain exhibits an exceptionally diverse range of cell types compared to other tissues <sup>9,10</sup>. As a result, single-cell RNA sequencing in the brain is crucial for gaining deeper insights into brain function within various cellular contexts. Due to the highly interconnected nature of brain tissue, isolating nuclei for RNA sequencing—known as single-nucleus RNA sequencing (snRNA-seq)—is a more commonly used approach in brain research than scRNA-seq. Recent studies utilizing snRNA-seq in neurodegenerative diseases, such as Parkinson's disease and Alzheimer's disease, have uncovered disease-vulnerable subtypes of neurons and associated glial cell subtypes, shedding light on the heterogeneity and complexity of the cellular landscape in neurodegenerative disorders <sup>11–15</sup>.

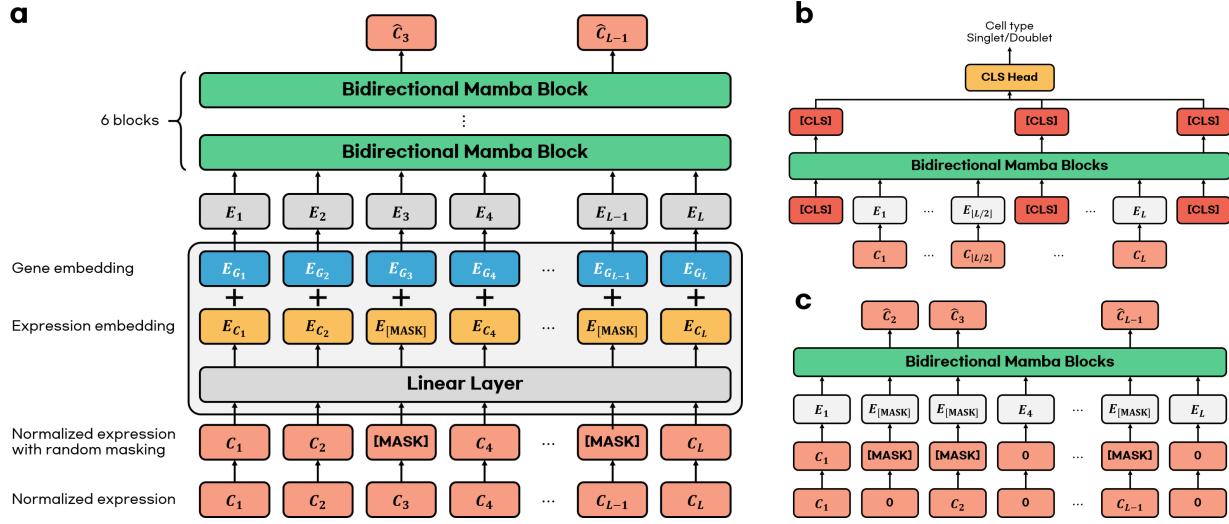
However, snRNA-seq analysis in neurodegenerative diseases faces several significant challenges. First, the low quality of postmortem brain samples, which are obtained from brains at varying postmortem intervals, often results in poor-quality snRNA-seq data. Second, the high variability of snRNA-seq, compounded by disease heterogeneity, makes it even more difficult to integrate data from multiple sources. Finally, the limited quantity of mRNA within a single nucleus increases the likelihood of gene expression failure, a phenomenon known as “dropout”. As a result, snRNA-seq data often contains a high number of zero counts, making it essential to differentiate between true biological zeros and false zeros caused by technical noise.

To address these challenges, several computational approaches have been developed, primarily targeting single-cell RNA sequencing. However, two critical issues remain in previously developed methods. First, numerous imputation techniques have been introduced to resolve dropout events in scRNA-seq data<sup>16–20</sup>. However, many of these existing methods suffer from long computational runtimes, underscoring the need for a more efficient approach to impute missing values in scRNA-seq data.

Another challenge stems from the long sequence length of snRNA-seq data. Typically, snRNA-seq captures the expression levels of tens of thousands of genes, and cell type annotation methods<sup>3,4,21,22</sup> classify cell types based on gene expression patterns. However, analyzing information from all genes is computationally demanding and often exceeds the capacity of many models. To address this, many annotation methods focus on highly variable genes (HVGs), a subset of a few thousand genes, and rely exclusively on their expression levels. Unfortunately, the selection of HVGs is sensitive to parameter settings and can vary significantly across datasets and batches (e.g., between different patients). Moreover, selecting too few HVGs risks losing critical cellular information. This underscores the need for analysis methods capable of utilizing information from all genes without relying on HVG selection.

In recent years, pre-trained models have gained significant attention and been applied across various data types<sup>23–26</sup>. These models typically undergo pre-training using large, unlabeled datasets via self-supervised learning to extract generalizable features within a specific data domain. After pre-training, these foundation models can be fine-tuned with smaller labeled datasets, enabling their effective application to a wide range of downstream tasks.

Inspired by the success of pre-trained models, several pre-trained models for scRNA-seq have been proposed<sup>22,27–29</sup>. Some of these methods utilize Transformer architectures, where com-



**Fig. 1: Overall framework of scMamba.** **a.** The scMamba model comprises a linear layer for expression embeddings, gene embeddings, and bidirectional Mamba blocks. During pre-training, a subset of input data is masked, and the model predicts the expression levels at the masked positions. **b.** For fine-tuning classification tasks, three [CLS] embeddings are inserted into the input embeddings. These embeddings are processed through the Mamba blocks and then passed to a classification head, which predicts cell classes. **c.** For fine-tuning the snRNA-seq imputation task, portions of the input expression levels are masked, and the model predicts the masked values. Zero and non-zero values are masked with different probabilities to ensure balanced learning. Both zero and non-zero values are masked with different masking probabilities.

putational complexity increases quadratically with the input length. To address this, they often rely on a subset of genes to reduce computational demands. Other methods discretize expression values into bins, treating scRNA-seq data similarly to tokens in language models. While these strategies effectively manage processing requirements, they risk introducing significant information loss in the scRNA-seq data.

Recently, a novel architecture called Mamba<sup>30</sup> was introduced. Mamba is based on selective state space models (SSMs), enabling it to select data in an input-dependent manner dynamically. It also offers lower computational complexity than Transformers with self-attention, allowing for faster processing of long sequences. Mamba has demonstrated superior performance over Transformers and other SSM-based architectures in specific tasks and has been successfully applied

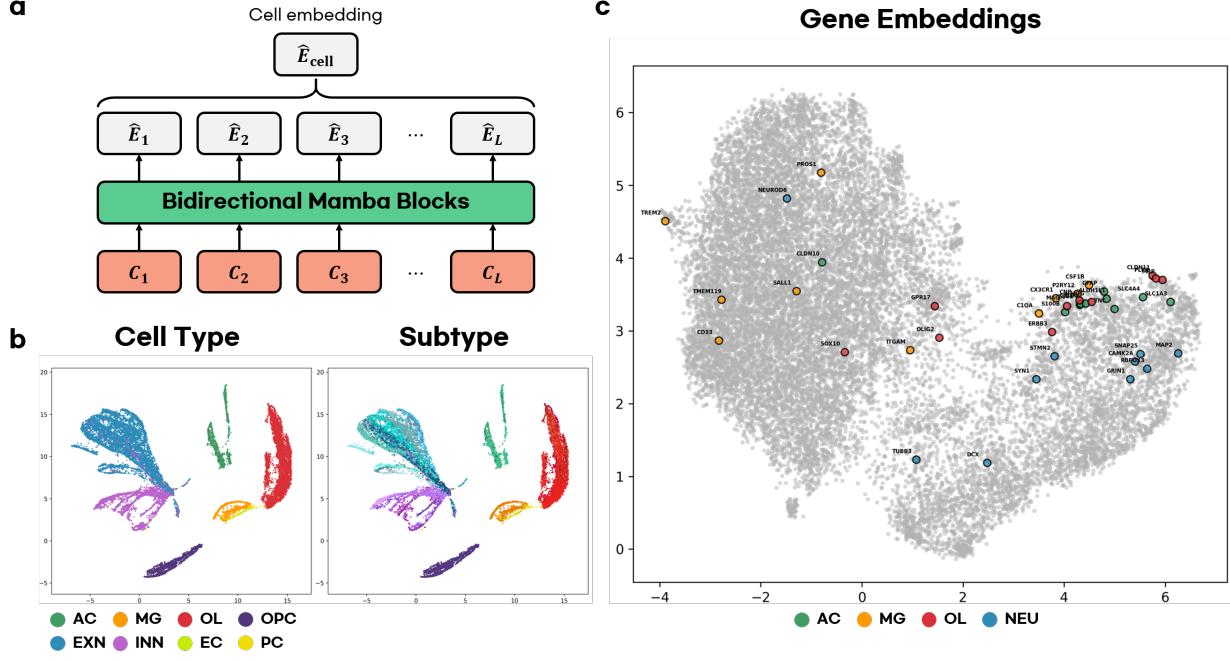
across various domains, achieving strong results<sup>31–36</sup>.

Inspired by this, we propose scMamba (Fig. 1a), an enhanced pre-trained model for analyzing brain snRNA-seq data. scMamba incorporates the Mamba block instead of self-attention, enabling it to process long snRNA-seq data without requiring dimensionality reduction. By pre-training the model using masked expression modeling, we demonstrate its effectiveness in downstream tasks such as cell type classification, doublet detection, and snRNA-seq imputation (Fig. 1b, c). Moreover, we show that scMamba consistently outperforms comparative methods across five diverse datasets from different brain tissues.

## Results

**scMamba learn meaningful representation during pre-training.** To validate that scMamba learns meaningful cell representations during pre-training, we extracted cell features from the pre-trained model and visualized them in 2D space using UMAP<sup>37</sup>. Specifically, we obtained the output of the pre-trained model with dimensions  $L \times D$ , where  $L$  represents the length of snRNA-seq data and  $D$  denotes the embedding dimension of the model. We then calculated the average along the length dimension, which yielded cell embeddings of dimension  $D$  for each cell (Fig. 2a).

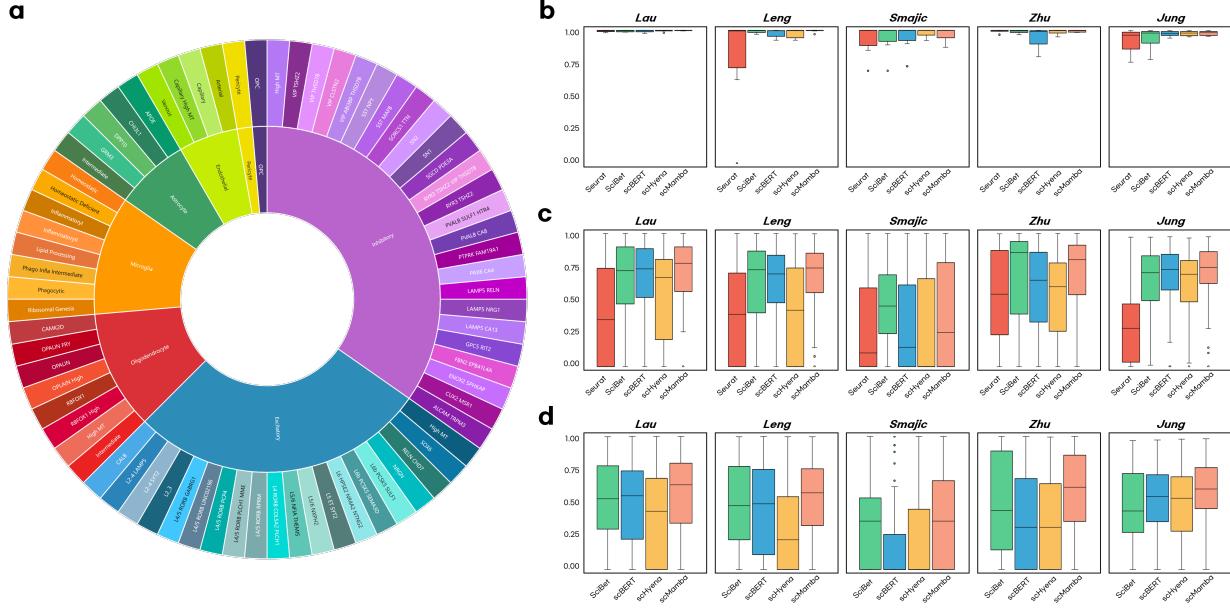
Fig. 2b presents UMAP visualizations of cell embeddings from the *Lau* dataset. Notably, the cell embeddings generated by scMamba form distinct clusters corresponding to cell types, despite the absence of cell type labels during pre-training. Furthermore, cells belonging to the same subtypes are predominantly grouped within these clusters. The results of other datasets can be found in Supplementary Fig. 1. This highlights the ability of scMamba to capture meaningful cell representations during pre-training.



**Fig. 2:** a. To generate cell embeddings from the pre-trained model, snRNA-seq data is input into the model, and the resulting output features are averaged along the sequence length. b. UMAP visualization of cell embeddings from the pre-trained scMamba model. Each UMAP is colored based on 8 major cell types or 72 subtypes. c. UMAP visualization of gene embeddings from pre-trained scMamba model. Marker genes of 4 distinct cell types are labeled with names. (AC: astrocyte, MG: microglia, OL: oligodendrocyte, OPC: oligodendrocyte progenitor cell, EXN: excitatory neuron, INN: inhibitory neuron, EC: endothelial cell, PC: pericyte, NEU: neuron).

Next, we validate that the gene embeddings learned by the pre-trained scMamba model capture meaningful information. Fig. 2c shows a UMAP visualization of the gene embeddings learned by scMamba, as illustrated in Fig. 1a. In this visualization, the embeddings of marker genes associated with specific cell types are positioned near one another. This indicates that scMamba successfully learns relationships among genes through its gene embeddings during pre-training.

**scMamba is capable of classifying sub cell types.** For the cell type classification tasks, our dataset was categorized into 8 major cell types. Additionally, the dataset was further subdivided into 72 subtypes and 127 fine-grained subclusters. Fig. 3a provides an overview of the cell types and subtypes included in our dataset. We conducted cell type classification across three hierarchi-



**Fig. 3:** a. The dataset is labeled with 8 major cell types and 72 detailed subtypes. b. F1 score distribution across 8 cell types, with each box plot representing results for individual datasets. c. F1 score distribution across 72 subtypes, with each box plot representing results for individual datasets. d. F1 score distribution across 127 subclusters, with each box plot representing results for individual datasets.

cal levels.

To evaluate the performance of scMamba in the cell type classification task, we performed a comparative analysis against four baseline methods: Seurat<sup>4</sup>, SciBet<sup>3</sup>, scBERT<sup>22</sup>, and scHyena<sup>38</sup>. First, Seurat, a widely used tool for single-cell analysis, was employed. Using Seurat, we clustered the snRNA-seq data and manually assigned cell types to each cluster by comparing the marker genes of the clusters with known marker genes for each cell type. Next, SciBet, a supervised cell type annotation method specifically designed for scRNA-seq data, was trained using the same training set utilized for fine-tuning scMamba. scBERT, another baseline, is a pre-trained model designed for cell type annotation of scRNA-seq data, based on the Performer architecture<sup>39</sup>. A key difference between scBERT and scMamba is that scBERT discretizes expression levels into bins, while scMamba processes expression levels in their continuous form. Finally, we included

scHyena as a baseline method. scHyena is another pre-trained model for snRNA-seq data analysis using Hyena <sup>40</sup> operator instead of self-attention. For a fair comparison, we modified scHyena to use three [CLS] embeddings, as in scMamba, even though the original scHyena implementation uses only a single prepended [CLS] embedding. To ensure the reproducibility of results on our datasets, we referred to the official source codes of these baseline methods. Additionally, scBERT and scHyena were pre-trained on the same dataset used for pre-training scMamba.

Fig. 3b illustrates the F1 score distribution across 8 major cell types. Both scHyena and scMamba achieve consistently high F1 scores across all cell types in each dataset. Moreover, as shown in Supplementary Table 4, scHyena and scMamba generally achieve the highest average F1 scores across most experiments. However, other baseline methods also demonstrate strong performance, indicating that classifying major cell types is a relatively straightforward task for classification methods.

To evaluate the classification methods more comprehensively, we applied them to subtype and subcluster classification tasks. Fig. 3c depicts the F1 score distributions for each classification method. As shown, Seurat consistently exhibits the lowest performance across all datasets. Since Seurat assigns clusters to cell types by manually comparing the marker genes of clusters and cell types, its accuracy is highly dependent on the selection of marker genes. Furthermore, the manual nature of Seurat’s classification process demands significant time and effort, which can also affect its overall accuracy.

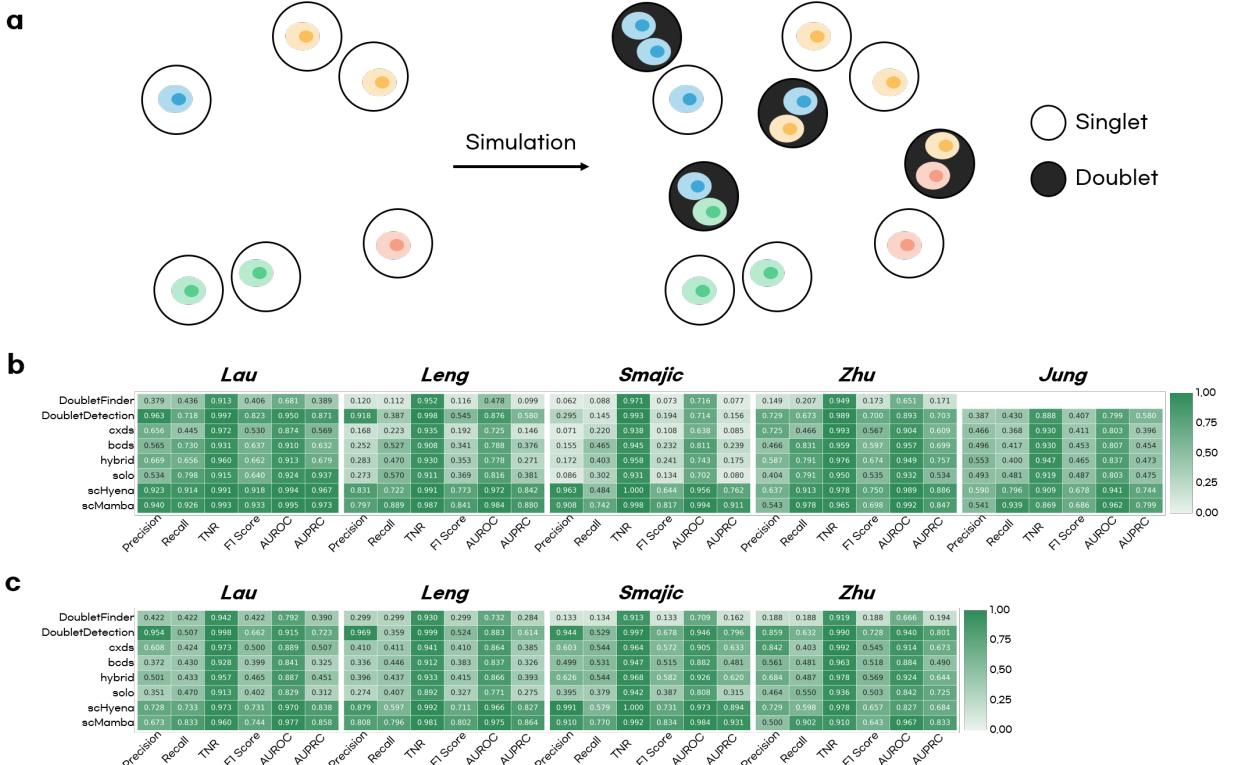
SciBet performs relatively strongly in subtype classification but falls short compared to the proposed method. scBERT performs well in most cases, however, its performance declines with the *Smajic* and *Zhu* datasets. While scHyena excels in major cell type classification, its performance is more constrained in subtype classification, highlighting certain limitations in identifying detailed

cell types.

In contrast, the proposed scMamba consistently delivers high performance across all datasets, as confirmed by Supplementary Table 4. Moreover, both Fig. 3d and Supplementary Table 4 demonstrate that scMamba achieves the highest F1 scores in subcluster classification. These results highlight the superior ability of scMamba to classify detailed cell types more accurately than other methods.

**scMamba effectively identifies and filters doublets.** During data preparation, we annotated doublets using Scrublet<sup>41</sup> to establish ground truth for doublet detection. However, the labels generated by Scrublet may not always be accurate. To ensure a fairer comparison, we also conducted experiments using simulated doublets. As illustrated in Fig. 4a, simulated doublets were created by randomly selecting two singlets and averaging their UMI counts. The number of simulated doublets was set to 10% of the total number of singlets. Experiments using Scrublet-annotated doublets are referred to as ‘in vivo’ doublet detection, while those using simulated doublets are referred to as ‘simulated’ doublet detection.

We followed the approach outlined in the previous study<sup>42</sup> to evaluate doublet detection performance. For baseline comparisons, we utilized seven methods, including scHyena: DoubletFinder<sup>43</sup>, DoubletDetection<sup>44</sup>, cxds, bcds, and hybrid from the scds<sup>45</sup>, as well as solo<sup>46</sup>. All baseline methods were implemented using their official source code. To assess doublet detection performance, we employed six widely used metrics: precision, recall, true negative rate (TNR), F1 score, area under the ROC curve (AUROC), and area under the precision-recall curve (AUPRC). Additionally, following the approach described in the previous study<sup>42</sup>, we evaluated doublet detection performance under a predefined percentage of droplets identified as doublets (identification rate).



**Fig. 4:** a. Simulated doublets are generated by averaging the UMI counts of two randomly selected singlets. b. Heatmap of evaluation metric scores for *in vivo* doublet detection by each method across datasets. White squares indicate where the method failed to execute. c. Heatmap of evaluation metric scores for simulated doublet detection by each method across datasets.

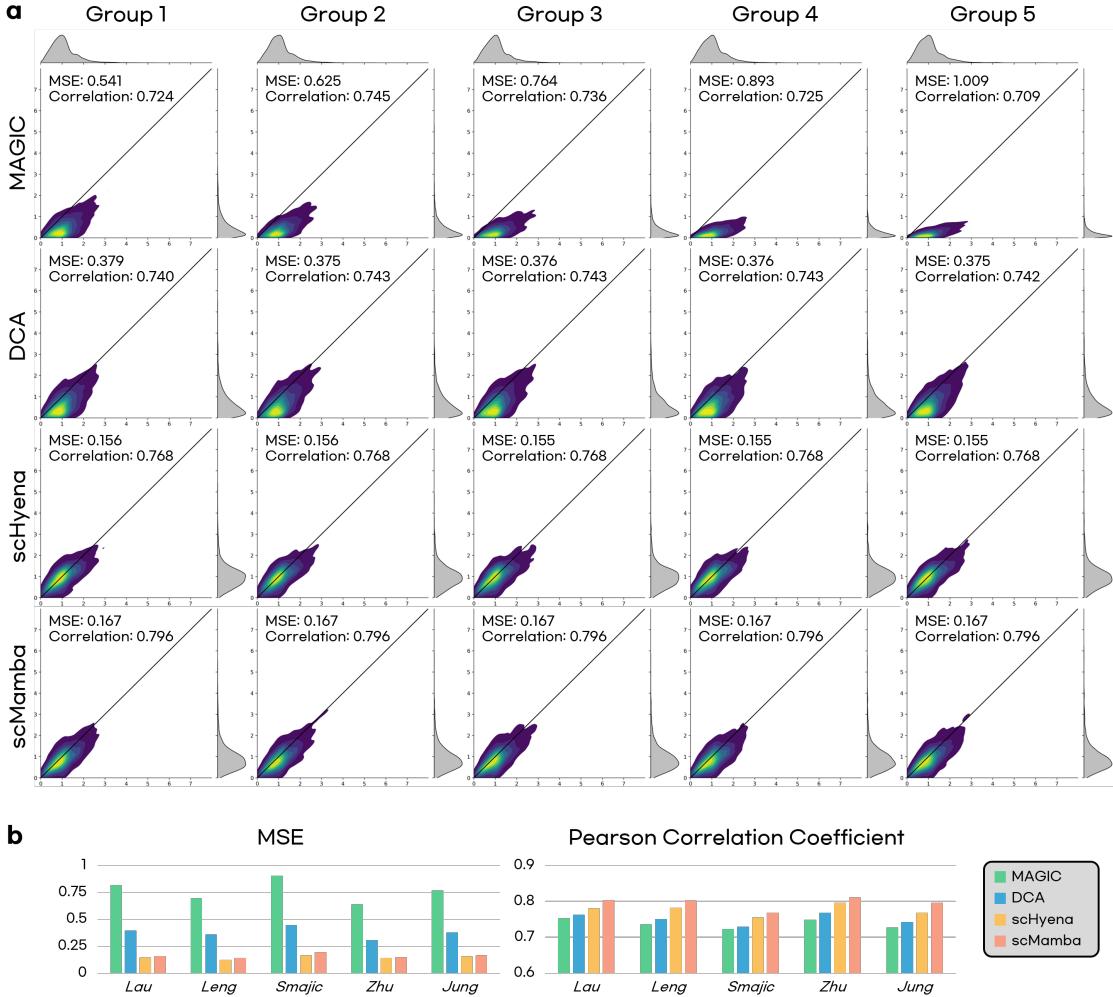
Fig. 4b, c present the evaluation metric scores of each method across datasets for *in vivo* and simulated doublet detection, respectively. DoubletFinder consistently demonstrates the lowest performance, particularly with low precision, recall, and F1 scores across all experiments. Its high computational demands render it unsuitable for larger datasets, such as the *Jung* dataset. Doublet-Detection generally achieves high precision and TNR, effectively identifying doublets while minimizing false positives (i.e., singlets misclassified as doublets). However, it does not surpass the proposed method in other metrics. While cxds performs poorly in *in vivo* experiments but improves with simulated data, bcds shows the opposite trend. Despite these variations, both methods exhibit low overall metric scores. Similarly, the hybrid method, which combines cxds and bcds, delivers

performance comparable to its component methods, while solo achieves slightly lower results. In contrast, scHyena and scMamba consistently deliver strong performance in doublet detection. Notably, scMamba achieves the highest recall, F1 score, AUROC, and AUPRC in most experiments. These results confirm that scMamba is highly effective for accurate doublet detection.

Supplementary Fig. 2 and 3 present evaluation metric scores for in vivo and simulated doublet detection, respectively. Across most experiments with specific identification rates, scMamba consistently achieves the highest precision, recall, and TNR. These findings further demonstrate the robust doublet detection capability of the proposed method.

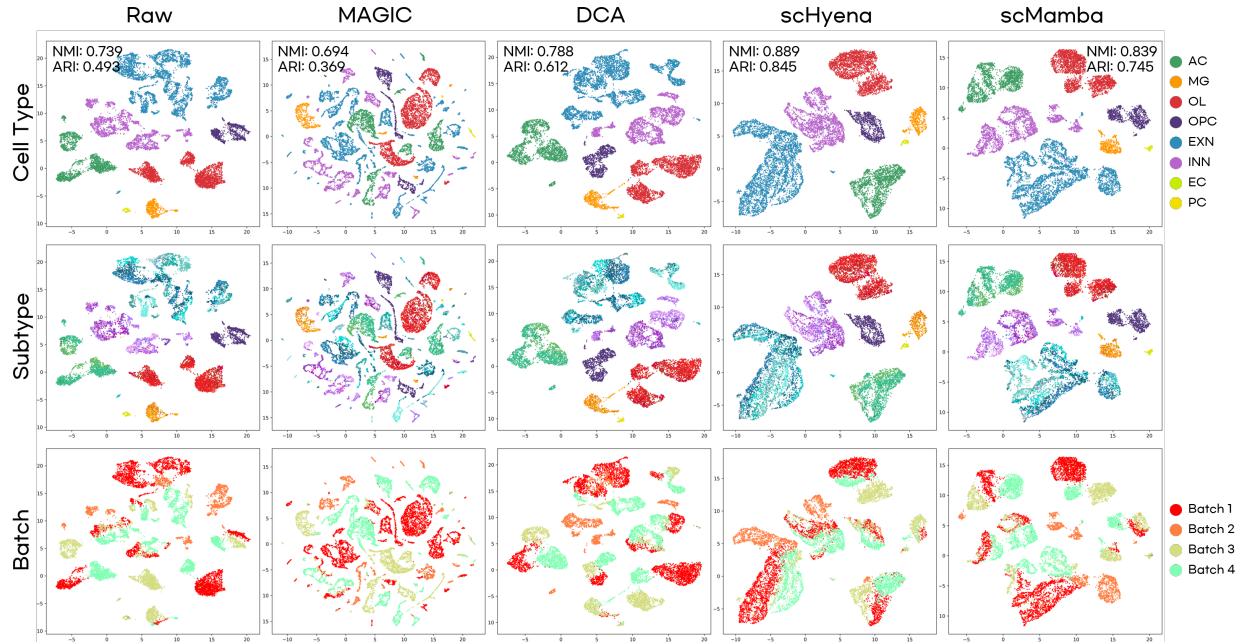
**scMamba can impute snRNA-seq data and correct batch effects.** To evaluate the imputation performance of scMamba, we compared it with baseline methods MAGIC<sup>18</sup>, DCA<sup>20</sup>, and scHyena. For a quantitative assessment, we first employed a ‘nonzero imputation’ approach: nonzero values in the input data were masked, and the imputation methods were used to predict these masked values. To ensure comprehensive evaluation, we divided the nonzero indices into five subgroups and evaluated the imputation methods independently on each subgroup. Performance was measured using the Mean Squared Error (MSE) and Pearson correlation coefficient between the true and imputed values at the masked indices. Only data with a UMI count greater than 4,000 were included for accurate evaluation.

Fig. 5a displays joint plots comparing true and imputed values, along with MSEs and Pearson correlation coefficients for each subgroup in the *Jung* dataset. The results show that scHyena and scMamba outperform MAGIC and DCA methods in both MSE and Pearson correlation coefficients. Specifically, MAGIC and DCA tend to underestimate values relative to the true values. In contrast, the joint plots for scMamba reveal that most points closely align with the  $y = x$  line, indicating a strong agreement between true and imputed values. These results demonstrate that



**Fig. 5:** Results of nonzero imputation experiments. **a.** Joint plots comparing true and imputed values predicted by various imputation methods on the *Jung* dataset. The x and y-axis represent true values and imputed values, respectively. Values in the upper left corner of each joint plot indicate the MSEs and Pearson correlation coefficients. **b.** MSE and Pearson correlation coefficient for nonzero imputation by each method across datasets. The histogram values represent the averages of the five groups.

scMamba provides more accurate imputations with significantly lower errors than MAGIC and DCA. Fig. 5b presents histograms of the average MSE and Pearson correlation coefficient for nonzero imputation by each method across datasets. Once again, scHyena and scMamba exhibit low MSEs and high Pearson correlation coefficients, highlighting their superior performance. In contrast, MAGIC and DCA demonstrate comparatively poorer results, indicating their limitations



**Fig. 6: Results of zero imputation experiments. UMAP visualizations of raw and imputed snRNA-seq data for the *Leng* dataset. The figures are labeled with cell type, subtype, and batch (patient), respectively.**

in imputation tasks.

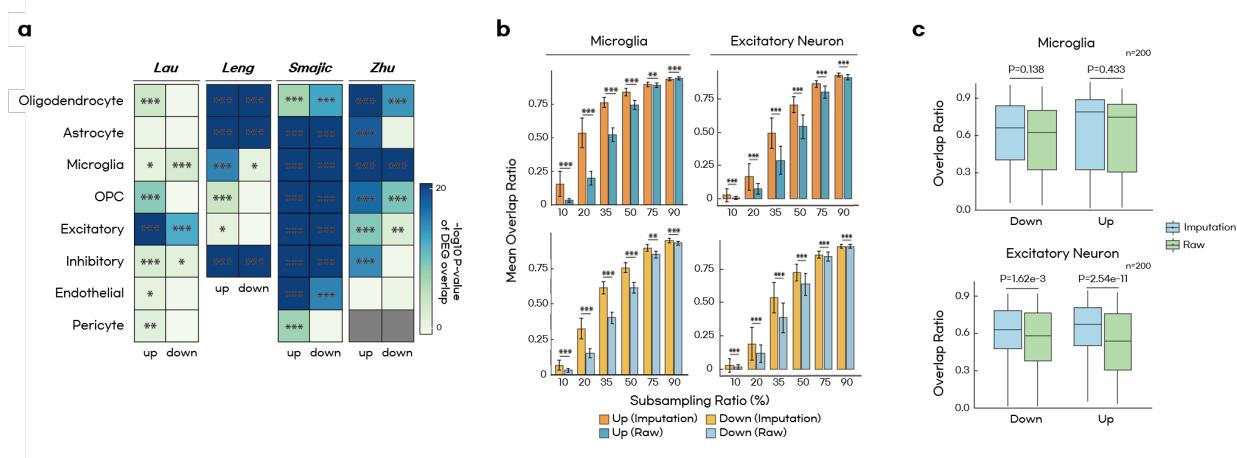
To further evaluate the imputation performance of scMamba, we conducted a ‘zero imputation’ analysis, where zero values in the snRNA-seq data were imputed using various methods and results were visualized in 2D space using UMAP. Theoretically, accurate imputation of zero values should lead to denser clusters, grouping samples more distinctly with others of the same cell type. As a result, imputation can help reduce batch effects in the data. To quantitatively evaluate batch effect reduction, we measured normalized mutual information (NMI) and adjusted Rand index (ARI), both of which are commonly used to assess clustering performance.

Fig. 6 displays UMAP plots of raw and imputed snRNA-seq data for the *Leng* dataset. The first column shows the UMAP of raw counts (before imputation), where cells of the same type are clustered together, but cells from different batches (patients) are noticeably separated. This pattern

suggests that the data is influenced more by technical factors than by biological characteristics. When MAGIC is applied for imputation, clusters become scattered, and the batch effect remains uncorrected. Furthermore, NMI and ARI values decrease compared to the raw counts, indicating poorer clustering performance. Similarly, DCA fails to correct the batch effects, and it introduces an unintended connection between clusters of microglia and oligodendrocytes within the same patient group, implying inaccuracies in its imputed values. In contrast, snRNA-seq data imputed with scHyena forms dense clusters of cells belonging to the same brain cell type. scHyena also achieves the highest NMI and ARI metric values, reflecting its strong performance in reducing batch effects. Likewise, scMamba effectively reduces batch effects, as demonstrated by the UMAP visualization and the quantitative metrics. These findings confirm that scMamba can impute snRNA-seq data with biologically meaningful values, thereby mitigating batch effects and enhancing data quality.

**scMamba improves robustness in DEG analysis.** To assess the impact of scMamba on imputation, we focused on differential gene expression (DEG) analysis. DEGs between diseased and neurotypical normal samples were identified with and without imputation, using data from two Alzheimer's disease studies and two Parkinson's disease studies, using MAST (v1.32.0). Differential expression analysis was performed for eight major cell types using the FindMarkers pipeline in Seurat, with default parameters (adjusted P-value (FDR)  $\leq 0.05$ , log2FC  $\geq 0.25$ ) in R. The performance of scMamba imputation was evaluated by determining the fraction of overlapping DEGs between subsampled data and the original 100% dataset across 100 permutations for each cell type in individual studies. Subsampling was performed randomly in each round from the original dataset. DEGs were categorized into upregulated and downregulated groups, which were then matched against the corresponding DEGs identified in each subsampled dataset.

In Fig. 7a, scMamba imputation with 50% subsampling effectively recovered differentially expressed genes (DEGs) from the original dataset across most cell types and studies, outperforming



**Fig. 7: Evaluating scMamba performance in single-cell differential expression analysis.** **a.** The overlap ratio represents the intersection of DEGs between 50% subsampled data and the original 100% dataset. -log10 P-values were calculated using a paired t-test for each overlap ratio across 100 permutations. The grey box indicates missing values due to insufficient overlap to calculate P-values. The -log10 P-value range is capped at a maximum of 20 and a minimum of 0. Statistical significance is denoted as follows: \*\*\* for  $p \leq 0.001$ , \*\* for  $p \leq 0.01$ , and \* for  $p \leq 0.05$ . **b.** Assessment of scMamba robustness through multiple subsampling ratios, showing the mean overlap ratio from 100 permutations for microglia (3717 cells) and excitatory neurons (980 cells) in the *Smajic* dataset. Error bars indicate standard deviation of each subsampled value. P-values were calculated using a paired t-test based on 100 permutations of the overlap ratio. Statistical significance is denoted as follows: \*\*\* for  $p \leq 0.001$  and \*\* for  $p \leq 0.01$ . **c.** Boxplots showing the reproducibility of DEGs across all possible combinations ( $n=200$ ) of half of the patients, compared to the original data, with and without imputation. Specifically, from a total of 5 patients and 6 neurotypical normal samples, we selected 3 patients and 3 neurotypical normal samples. Statistical significance was tested using a paired t-test.

the results obtained without imputation. Despite variations across cell types and studies, both upregulated and downregulated genes showed statistically significant improvements in the recovery rate of DEGs after imputation. To further investigate the impact of scMamba imputation on various subsampled ratios, we specifically focused on microglia and excitatory neurons from the *Smajic* dataset (ref), selecting one large population and one smaller cell population, respectively. Across varying subsampling ratios, scMamba imputation consistently retained more DEGs than the non-imputed data (Fig. 7b). For example, with only 20% subsampling, scMamba imputation recovered about 50% of the original upregulated DEGs in microglia, while the non-imputed data recovered only 20%. Additionally, we compared the overlap ratio of DEGs by randomly selecting half of

the samples from the same study, with and without imputation. As expected, scMamba imputation enhanced reproducibility across all cell types, with a more significant impact in datasets with lower cell numbers (Fig. 7c and Supplementary Fig. 4).

Overall, these results demonstrate that scMamba enhances both the quality and reproducibility of data, which is crucial for unbiased integration, especially when addressing the high variability introduced by low-quality postmortem brain tissue and the heterogeneity of disease.

## Discussion

In this work, we introduced scMamba, a pre-trained model designed to enhance the quality and utility of snRNA-seq analysis, particularly in studies of neurodegenerative diseases. Built upon the Mamba model, scMamba can directly process raw snRNA-seq data without the need for dimensionality reduction. We demonstrated that our model effectively learns meaningful representations of cells and genes through pre-training using masked expression modeling, even without incorporating additional information during pre-training. As a result of these learned representations, scMamba achieved outstanding performance across various downstream tasks, including cell type classification, doublet detection, and snRNA-seq imputation.

In cell type classification, Seurat demonstrates lower performance than other methods at the subtype and subcluster levels. As cell types are further subdivided, it becomes increasingly challenging to assign unique marker genes to each type. Moreover, fine-grained clustering requires matching all clusters to specific types, a process that is not only time-consuming but also prone to reducing classification accuracy.

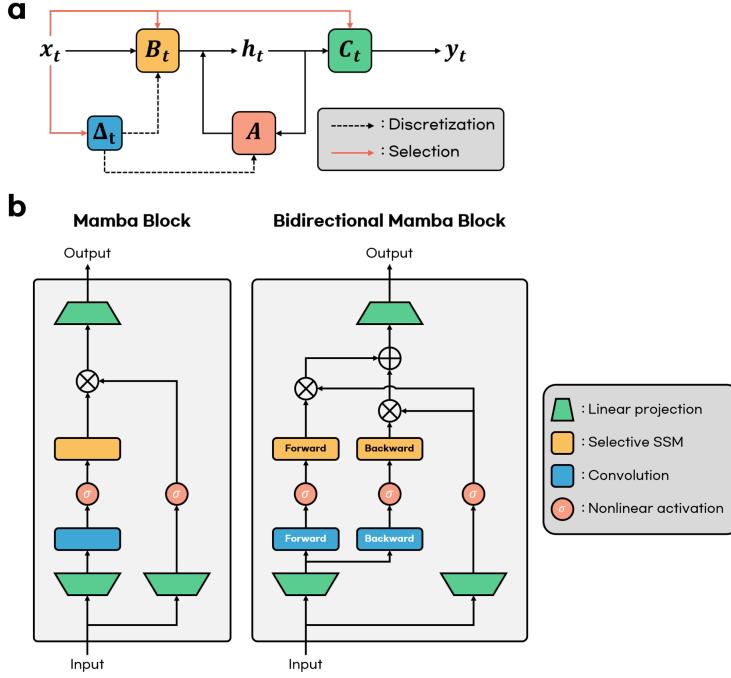
scBERT achieves strong classification performance but does not outperform the proposed

method, possibly due to differences in data handling. scBERT discretizes inputs by binning, which can lead to information loss. In contrast, scMamba processes inputs directly without binning or discretization, allowing it to retain more information and outperform scBERT.

scHyena performs well in major cell type classification but exhibits diminished accuracy in subtype and subcluster classification, indicating a limited ability to distinguish finer cell types. In contrast, scMamba consistently achieves superior performance across all classification levels. We hypothesize that the ability of Mamba to selectively retain information based on inputs contributes to the superior performance of scMamba.

In the doublet detection experiments, DoubletDetection outperforms scMamba in precision and TNR, while scMamba demonstrates stronger performance across other metrics. This indicates that scMamba identifies more doublets overall, including some additional false positives compared to DoubletDetection. However, because thorough removal of doublets is critical, detecting doublets with high sensitivity is advantageous, even if it results in a slightly higher false positive rate, as long as other metrics remain robust. Furthermore, we confirmed that scMamba surpasses DoubletDetection in precision and TNR at a fixed doublet identification rate. From this perspective, scMamba offers a more effective approach than DoubletDetection for accurate doublet detection.

Although snRNA-seq is an advanced technology, analyzing the underlying causes of neurodegenerative disorders using snRNA-seq remains challenging because these datasets are typically derived from postmortem brain samples. Variations in postmortem intervals can degrade sample quality, further complicating snRNA-seq analysis. Additionally, disease heterogeneity poses significant obstacles to integrating snRNA-seq data from multiple sources. To assess these challenges, accurate imputation of snRNA-seq data is crucial. In our experiments, we demonstrated that scMamba accurately imputes snRNA-seq data, effectively correcting batch effects. Further-



**Fig. 8: State space models (SSMs) and Mamba.** **a.** Selective SSMs dynamically adjust their parameters based on input data, enabling selective retention of information relevant to the input. It allows them to retain information based on input selectively. **b.** Selective state space models (SSMs) are organized into modular Mamba blocks, with the Mamba model being constructed by stacking multiple Mamba blocks.

more, we found that imputation using scMamba enhances the robustness of differential gene expression (DEG) analysis. These findings suggest that scMamba can facilitate the integration of snRNA-seq data from various sources, supporting large-scale studies. Moreover, they highlight the potential scalability of scMamba for analyzing snRNA-seq data to uncover the causes of neurodegenerative disorders.

## Methods

**State space models.** State space models (SSMs) have been widely utilized in control theory. Recently, SSMs have gained attention in deep learning for sequence data, as they can map input sequences to output sequences through a latent state. In deep learning applications, where input

data are typically discrete signals, the discretized form of SSMs is adopted. Discrete SSMs can be expressed as follows:

$$\begin{aligned} h_t &= \bar{\mathbf{A}}h_{t-1} + \bar{\mathbf{B}}x_t, \\ y_t &= \mathbf{C}h_t, \end{aligned} \tag{1}$$

where  $\bar{\mathbf{A}}$ ,  $\bar{\mathbf{B}}$ , and  $\mathbf{C}$  are the parameters of SSM. Here,  $\bar{\mathbf{A}}$  and  $\bar{\mathbf{B}}$  are the discretized form of  $\mathbf{A}$  and  $\mathbf{B}$ , defined as:

$$\begin{aligned} \bar{\mathbf{A}} &= \exp(\Delta\mathbf{A}), \\ \bar{\mathbf{B}} &= (\Delta\mathbf{A})^{-1}(\exp(\Delta\mathbf{A}) - \mathbf{I} \cdot \Delta\mathbf{B}), \end{aligned} \tag{2}$$

where  $\Delta$  represents the learnable step size parameter.

While the self-attention mechanism in Transformer <sup>47</sup> explicitly calculates relationships between inputs, SSMs implicitly capture these relationships through compressed state variables. This allows SSMs to handle longer sequences with significantly lower computational complexity than Transformers. However, unlike self-attention maps, which are dynamically computed based on the input, the parameters of SSMs remain static and do not adapt to the input. This lack of input-dependent adaptability limits the performance of SSMs in certain scenarios.

The selective SSM (S6) <sup>30</sup> was introduced to address the limitation of standard SSMs. As illustrated in Fig. 8a, S6 allows  $\mathbf{B}$ ,  $\mathbf{C}$ , and  $\Delta$  to become input-dependent as follows:

$$\begin{aligned} \mathbf{B} &= \mathbf{Linear}_N(x), \\ \mathbf{C} &= \mathbf{Linear}_N(x), \\ \Delta &= \log(1 + \exp(\mathbf{Parameter} + \mathbf{Broadcast}_D(\mathbf{Linear}_1(x)))). \end{aligned}$$

This adaptation enables SSMs to be content-aware, allowing them to selectively retain or discard information based on input.

**Mamba.** Selective SSMs can be organized into modular blocks. Recently, the Mamba block <sup>30</sup> was introduced, which utilizes selective SSMs, as shown in Fig. 8b. By stacking multiple Mamba blocks, a structure resembling the Transformer decoder can be constructed.

However, since SSMs are causal systems, they are inherently limited in their application to non-temporal data, such as visual data. To address this limitation, some studies propose bidirectional Mamba blocks <sup>32,35</sup>, and Fig. 8b illustrates an example of a bidirectional Mamba block.

**scMamba.** The proposed scMamba model is illustrated in Fig. 1a. To effectively process long snRNA-seq sequences, we adopted the Mamba architecture composed of multiple Mamba blocks. In the context of snRNA-seq data, the concept of time causality does not apply, as any gene can potentially relate to others regardless of its position in the sequence. To account for this, we implemented a non-causal bidirectional Mamba block, enabling the output to depend on the entire input across all sequence positions.

An input snRNA-seq consists of the normalized expression levels of  $L$  genes, denoted as  $(C_1, C_2, \dots, C_L)$ . Unlike natural language, where words are tokenized into discrete tokens and each token is mapped to a unique embedding, gene expression levels are continuous values that cannot be directly discretized. A previous method <sup>22</sup> addressed this issue by discretizing expression values into bins. However, this approach risks significant information loss in the snRNA-seq data. To mitigate this concern, we encode the expression levels into expression embeddings  $(E_{C_1}, E_{C_2}, \dots, E_{C_L})$  using a linear adapter layer. This approach preserves the continuous nature of gene expression data, ensuring no loss of information during the encoding process.

Another key difference between language and snRNA-seq data is that the order of genes in snRNA-seq has no inherent meaning. Instead, it is crucial to provide information indicating which gene’s expression level corresponds to each position in the sequence. To achieve this, we incorporate gene embeddings into the scMamba model, replacing the positional encodings used in traditional Transformers. In this approach, each gene is assigned a unique embedding ( $E_{G_1}, E_{G_2}, \dots, E_{G_L}$ ), which is then added to the expression embeddings. This strategy ensures that the scMamba model receives explicit gene-related information, enabling it to effectively interpret the snRNA-seq data. After combining the gene embeddings with the expression embeddings, the final input embeddings ( $E_1, E_2, \dots, E_L$ ) are generated and fed into the Mamba blocks.

**Pre-training.** Fig. 1a illustrates the pre-training process of the scMamba model. To pre-train our model, we employ a technique called masked expression modeling (MEM), inspired by the concept of masked language modeling <sup>23</sup>. In this approach, a subset of input embeddings is randomly replaced with an [MASK] embedding, and scMamba is trained to predict the expression levels of the masked genes. The masking probability is set to 0.15, and only nonzero values are masked, as distinguishing between true and false zero values is not feasible.

Since genes can interact regardless of their positions, predicting masked expressions requires considering relationships between all genes. To achieve this, scMamba uses bidirectional Mamba blocks, enabling it to predict expression levels at masked positions. The pre-training objective is defined as:

$$\ell_{MEM} = \sum_{i \in M} (C_i - \hat{C}_i)^2, \quad (3)$$

where  $M$  denotes the set of masked indices, and  $C_i$  and  $\hat{C}'_i$  represent the true and predicted gene expression levels, respectively. This pre-training process allows scMamba to learn generalizable features of both cells and genes.

**Cell type classification and doublet detection.** Cell type classification and doublet detection are among the most important tasks in snRNA-seq analysis. Fig. 1b illustrates the fine-tuning process of scMamba for these tasks. To adapt the pre-trained scMamba model for cell type classification or doublet detection, we insert [CLS] embeddings into the input data embeddings. While the [CLS] token or embedding is typically prepended to the input, we observed that placing three [CLS] embeddings—at the beginning, middle, and end of the input—enhanced performance.

After the input embeddings pass through the Mamba blocks, the three [CLS] embeddings are concatenated and fed into a classification head. The output of the classification head generates logits, and scMamba is fine-tuned using the cross-entropy loss,  $\ell_{cls} = -\sum_{i=1}^{N_c} y_i \log p_i$ , where  $N_c$  is the number of classes,  $y_i$  is the true class label, and  $p_i$  is the Softmax probability derived from the output of scMamba. The hyperparameters for pre-training and fine-tuning can be found in Supplementary Table 1.

**snRNA-seq imputation.** Imputing missing values in snRNA-seq data is essential due to the high prevalence of zeroes resulting from dropout events. One strategy for imputation involves directly adapting the pre-training approach. However, in pre-training, zero values are not masked, which may lead the model to learn to replace true zero values with other values incorrectly. Alternatively, if zero values are masked at the same probability as non-zero values, the model may disproportionately predict zeroes, given that most values in snRNA-seq data are zeroes.

To mitigate this issue, we apply different masking probabilities for zero and non-zero values. Specifically, non-zero values are masked with a probability of 0.4, while zero values are masked with a lower probability of 0.04. This adjustment balances the masking process for zero and non-zero values.

Despite this improvement, a challenge persists in distinguishing true zero values from false

zeroes caused by dropout. In some cases, the model may impute false zeroes as true zeroes. Fortunately, our scMamba model leverages gene embeddings, enabling it to learn and incorporate the expression level tendencies of individual genes.

Fig. 1c depicts the fine-tuning process for snRNA-seq imputation. The loss function for the imputation task remains consistent with the pre-training loss function, as shown in Equation (3). To ensure high-quality training data and improve model performance, we only used data with a UMI count greater than 4,000 for fine-tuning.

### Data preparation.

**Collection of published human neurodegenerative brain snRNA-seq data.** For this study, 14 distinct published snRNA-seq processed datasets were collected as gene-by-cell count matrices, with the following identifiers: GSE140231<sup>48</sup>, GSE148822<sup>49</sup>, GSE178265<sup>50</sup>, GSE157827<sup>51</sup>, GSE147528<sup>14</sup>, GSE129308<sup>52</sup>, GSE174367<sup>53</sup>, GSE167494<sup>54</sup>, GSE157783<sup>15</sup>, GSE160936<sup>55</sup>, GSE184950<sup>56</sup>, GSE163577<sup>57</sup>, GSE188545<sup>58</sup>, and GSE202210<sup>59</sup>. In addition, we utilized a custom dataset containing approximately 500,000 cells, referred to as the *Jung* dataset.

To integrate the data based on consistent transcriptomic information, Ensembl stable gene IDs were used in place of gene symbols. All datasets were concatenated using the union of Ensembl IDs, resulting in 61,325 genes. Undetected genes in each cell were assigned a value of 0. Genes located on chromosome Y or lacking annotation in the GRCh38 version 108 GTF file were filtered out. Additionally, only genes detected in at least 0.5% of all cells—excluding the *Kamath* dataset—were retained, yielding 19,306 unique Ensembl IDs.

Among the data, *Lau*<sup>51</sup>, *Leng*<sup>14</sup>, *Smajic*<sup>15</sup>, *Zhu*<sup>59</sup>, and *Jung* datasets were selected for down-

stream task, while the remaining 10 datasets were used for pre-training scMamba. The pre-training dataset comprises approximately 1.6 million cells. For Alzheimer’s disease, *Lau* and *Leng* datasets were chosen, representing three distinct brain regions—the prefrontal cortex, caudal entorhinal cortex, and superior frontal gyrus—while preserving the large size of the nucleus. For Parkinson’s disease, *Smajic* and *Zhu* datasets were selected, covering the substantia nigra and frontal cortex, respectively. The *Jung* dataset includes cases of both Alzheimer’s disease and Parkinson’s disease and spans regions such as the prefrontal cortex, hippocampus, and substantia nigra. The structure and detailed information about the datasets are provided in Supplementary Table 2 and 3.

**Cell type annotation based on unsupervised clustering.** The collected count matrices were preprocessed following the canonical SCANPY analysis pipeline<sup>60</sup>. The resulting data were integrated with the public datasets based on a unified list of 19,306 genes. Patients with less than 200 cells were filtered out, leaving a total of 2,408,023 nuclei from 461 patients. To identify doublets arising from experimental artifacts in the single-cell technique, we applied Scrublet<sup>41</sup> to calculate doublet scores and predict doublets for each single cell. These doublet scores were used to annotate doublet-enriched clusters. Cell clustering into distinct cell types was performed using the top 2,000 highly variable genes (HVGs), selected based on analytic Pearson residuals<sup>61</sup>. These genes were utilized for principal component analysis (PCA) and clustering. Total UMI counts per cell were normalized to 50,000, log2-transformed with a pseudo-count of 1, and scaled using scanpy.pp.scale. PCA coordinates were then computed using scanpy.pp.pca with default parameters. To mitigate batch effects across patients, Harmony correction<sup>62</sup> was applied to the PCA coordinates. Neighborhoods for every single cell were calculated using scanpy.pp.neighbors with the parameter of 20 PCA components and 40 nearest neighbors. The data were further reduced to a two-dimensional plane using UMAP for visualization. Finally, Leiden clustering was performed at a resolution of 1.8, resulting in 69 distinct clusters.

To annotate cell types for each cluster, the expression levels of known marker genes for major human brain cell types were analyzed across the clusters. Using a panel of 40 distinct marker genes, 51 clusters were assigned to 11 cell types, including 8 major brain cell types: Oligodendrocytes (CLDN11, MBP), Astrocytes (AQP4, ALDH1L1), Microglia (C1QC, CSF1R), Endothelial cells (CLDN5, FLT1), Pericytes (PDGFRB), Oligodendrocyte progenitor cells (OPCs; PDGFRA, VCAN), Excitatory neurons (SYT1, SLC17A7, SLC17A6), and Inhibitory neurons (SYT1, GAD1, GAD2). Additionally, three subtypes were identified: Neuron subtype 1 (SYT1, SLC17A6, GAD2), Neuron subtype 2 (SYT1, but neither SLC17A6 nor GAD2), and Myeloid subtype 1 (GNLY, CD44).

One cluster, which did not exhibit a clear expression pattern for any marker genes, was labeled as “unidentified”. Clusters with an average doublet score exceeding 0.1 were classified as doublets. For the cell type classification task, only single cells annotated to one of the 8 major cell types were considered.

**Preprocessing.** As part of our preprocessing pipeline for snRNA-seq data, we first filtered out cells with a total gene expression level below 200. Next, we normalized the gene expression values by scaling the total expression count of each cell to 10,000. Finally, log normalization ( $\log(x + 1)$ ) was applied to generate the final preprocessed dataset.

**Single-Cell Differential Expression Analysis.** We identified DEGs from imputed and raw data across four studies using MAST (v1.32.0), a method specifically designed for single-cell data analysis. Differential expression analysis was performed for eight major cell types, comparing disease and normal cells, following the FindMarkers pipeline in Seurat with default parameters (adjusted P-value (FDR)  $\leq 0.05$ ,  $\log_{2}FC \geq 0.25$ ) in R. The efficiency of scMamba was evaluated by determining the fraction of overlapping DEGs between 50% subsampled data and the original

100% dataset across 100 permutations. Subsampling was performed randomly each round from the original data. DEGs from the original data were categorized into upregulated and downregulated groups, which were then matched against upregulated and downregulated DEGs identified in each 50% subsampled dataset. The overlap ratio was calculated as the intersection of DEGs between the subsampled and original data, divided by the total DEGs from the original data. The matching DEGs between subsampled and original datasets was summarized by calculating the mean overlap ratio. P-values were determined using a paired t-test and transformed into -log<sub>10</sub> P-values for easier visualization of significance. The mean overlap ratio of microglia and excitatory neurons presented in Fig. 7b represents the average of overlapping DEGs across 100 permutations at each subsampling ratio (10%, 20%, 35%, 50%, 75%, 90%). To evaluate the reproducibility between patients, we randomly selected subsets of samples, specifically 3 patient samples and 3 normal samples, from a total of 5 patient samples and 6 normal samples covering all possible combinations (n=200). Differential expression analysis was then conducted on subsets of the eight major cell types. The overlap ratio of DEGs was compared within datasets, where “with imputation” refers to comparisons between imputed data, and “without imputation” refers to comparisons between raw data. The significance of these ratios was assessed using a paired t-test.

## References

1. Saliba, A.-E., Westermann, A. J., Gorski, S. A. & Vogel, J. Single-cell RNA-seq: Advances and future challenges. *Nucleic acids research* **42**, 8845–8860 (2014).
2. Rood, J. E., Maartens, A., Hupalowska, A., Teichmann, S. A. & Regev, A. Impact of the human cell atlas on medicine. *Nature medicine* **28**, 2486–2496 (2022).

3. Li, C. *et al.* SciBet as a portable and fast single cell type identifier. *Nature communications* **11**, 1818 (2020).
4. Hao, Y. *et al.* Integrated analysis of multimodal single-cell data. *Cell* **184**, 3573–3587 (2021).
5. Villani, A.-C. *et al.* Single-cell RNA-seq reveals new types of human blood dendritic cells, monocytes, and progenitors. *Science* **356**, eaah4573 (2017).
6. Jaitin, D. A. *et al.* Massively parallel single-cell RNA-seq for marker-free decomposition of tissues into cell types. *Science* **343**, 776–779 (2014).
7. Papalexi, E. & Satija, R. Single-cell RNA sequencing to explore immune cell heterogeneity. *Nature Reviews Immunology* **18**, 35–45 (2018).
8. Kinker, G. S. *et al.* Pan-cancer single-cell RNA-seq identifies recurring programs of cellular heterogeneity. *Nature genetics* **52**, 1208–1218 (2020).
9. Saunders, A. *et al.* Molecular diversity and specializations among the cells of the adult mouse brain. *Cell* **174**, 1015–1030 (2018).
10. Hodge, R. D. *et al.* Conserved cell types with divergent features in human versus mouse cortex. *Nature* **573**, 61–68 (2019).
11. Keren-Shaul, H. *et al.* A unique microglia type associated with restricting development of Alzheimer’s disease. *Cell* **169**, 1276–1290 (2017).
12. Mathys, H. *et al.* Single-cell transcriptomic analysis of Alzheimer’s disease. *Nature* **570**, 332–337 (2019).
13. Habib, N. *et al.* Disease-associated astrocytes in Alzheimer’s disease and aging. *Nature neuroscience* **23**, 701–706 (2020).

14. Leng, K. *et al.* Molecular characterization of selectively vulnerable neurons in Alzheimer's disease. *Nature neuroscience* **24**, 276–287 (2021).
15. Smajić, S. *et al.* Single-cell sequencing of human midbrain reveals glial activation and a Parkinson-specific neuronal state. *Brain* **145**, 964–978 (2022).
16. Huang, M. *et al.* SAVER: Gene expression recovery for single-cell RNA sequencing. *Nature methods* **15**, 539–542 (2018).
17. Li, W. V. & Li, J. J. An accurate and robust imputation method scImpute for single-cell RNA-seq data. *Nature communications* **9**, 997 (2018).
18. Van Dijk, D. *et al.* Recovering gene interactions from single-cell data using data diffusion. *Cell* **174**, 716–729 (2018).
19. Arisdakessian, C., Poirion, O., Yunits, B., Zhu, X. & Garmire, L. X. DeepImpute: An accurate, fast, and scalable deep neural network method to impute single-cell RNA-seq data. *Genome biology* **20**, 1–14 (2019).
20. Eraslan, G., Simon, L. M., Mircea, M., Mueller, N. S. & Theis, F. J. Single-cell RNA-seq denoising using a deep count autoencoder. *Nature communications* **10**, 390 (2019).
21. Aran, D. *et al.* Reference-based analysis of lung single-cell sequencing reveals a transitional profibrotic macrophage. *Nature immunology* **20**, 163–172 (2019).
22. Yang, F. *et al.* scBERT as a large-scale pretrained deep language model for cell type annotation of single-cell RNA-seq data. *Nature Machine Intelligence* **4**, 852–866 (2022).
23. Devlin, J. Bert: Pre-training of deep bidirectional Transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

24. Brown, T. *et al.* Language models are few-shot learners. *Advances in neural information processing systems* **33**, 1877–1901 (2020).
25. Bommasani, R. *et al.* On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258* (2021).
26. Ramesh, A. *et al.* Zero-shot text-to-image generation. In *International conference on machine learning*, 8821–8831 (Pmlr, 2021).
27. Theodoris, C. V. *et al.* Transfer learning enables predictions in network biology. *Nature* **618**, 616–624 (2023).
28. Cui, H. *et al.* scgpt: toward building a foundation model for single-cell multi-omics using generative ai. *Nature Methods* 1–11 (2024).
29. Hao, M. *et al.* Large-scale foundation model on single-cell transcriptomics. *Nature Methods* 1–11 (2024).
30. Gu, A. & Dao, T. Mamba: Linear-time sequence modeling with selective state spaces. In *First Conference on Language Modeling* (2024).
31. Huang, J. *et al.* MambaMIR: An arbitrary-masked mamba for joint medical image reconstruction and uncertainty estimation. *arXiv preprint arXiv:2402.18451* (2024).
32. Liu, Y. *et al.* VMamba: Visual state space model 2024. *arXiv preprint arXiv:2401.10166* (2024).
33. Qiao, Y. *et al.* VL-Mamba: Exploring state space models for multimodal learning. *arXiv preprint arXiv:2403.13600* (2024).
34. Schiff, Y. *et al.* Caduceus: Bi-directional equivariant long-range DNA sequence modeling. In *First Workshop on Long-Context Foundation Models @ ICML 2024* (2024).

35. Zhu, L. *et al.* Vision Mamba: Efficient visual representation learning with bidirectional state space model. In *Forty-first International Conference on Machine Learning* (2024).
36. Guo, H. *et al.* MambaIR: A simple baseline for image restoration with state-space model. In *European Conference on Computer Vision*, 222–241 (Springer, 2025).
37. McInnes, L., Healy, J. & Melville, J. UMAP: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426* (2018).
38. Oh, G., Choi, B., Jung, I. & Ye, J. C. schyena: Foundation model for full-length single-cell rna-seq analysis in brain. *arXiv preprint arXiv:2310.02713* (2023).
39. Choromanski, K. M. *et al.* Rethinking attention with Performers. In *International Conference on Learning Representations* (2021).
40. Poli, M. *et al.* Hyena hierarchy: Towards larger convolutional language models. In *International Conference on Machine Learning*, 28043–28078 (PMLR, 2023).
41. Wolock, S. L., Lopez, R. & Klein, A. M. Scrublet: Computational identification of cell doublers in single-cell transcriptomic data. *Cell systems* **8**, 281–291 (2019).
42. Xi, N. M. & Li, J. J. Benchmarking computational doublet-detection methods for single-cell rna sequencing data. *Cell systems* **12**, 176–194 (2021).
43. McGinnis, C. S., Murrow, L. M. & Gartner, Z. J. DoubletFinder: Doublet detection in single-cell RNA sequencing data using artificial nearest neighbors. *Cell systems* **8**, 329–337 (2019).
44. Gayoso, A., Shor, J., Carr, A. J., Sharma, R. & Pe'er, D. DoubletDetection (version v3.0) (2020).
45. Bais, A. S. & Kostka, D. scds: Computational annotation of doublets in single-cell RNA sequencing data. *Bioinformatics* **36**, 1150–1158 (2020).

46. Bernstein, N. J. *et al.* Solo: Doublet identification in single-cell RNA-seq via semi-supervised deep learning. *Cell systems* **11**, 95–101 (2020).
47. Vaswani, A. Attention is all you need. *Advances in Neural Information Processing Systems* (2017).
48. Agarwal, D. *et al.* A single-cell atlas of the human substantia nigra reveals cell-specific pathways associated with neurological disorders. *Nature communications* **11**, 4183 (2020).
49. Gerrits, E. *et al.* Distinct amyloid- $\beta$  and tau-associated microglia profiles in Alzheimer's disease. *Acta neuropathologica* **141**, 681–696 (2021).
50. Kamath, T. *et al.* Single-cell genomic profiling of human dopamine neurons identifies a population that selectively degenerates in Parkinson's disease. *Nature neuroscience* **25**, 588–595 (2022).
51. Lau, S.-F., Cao, H., Fu, A. K. & Ip, N. Y. Single-nucleus transcriptome analysis reveals dysregulation of angiogenic endothelial cells and neuroprotective glia in Alzheimer's disease. *Proceedings of the National Academy of Sciences* **117**, 25800–25809 (2020).
52. Otero-Garcia, M. *et al.* Molecular signatures underlying neurofibrillary tangle susceptibility in Alzheimer's disease. *Neuron* **110**, 2929–2948 (2022).
53. Morabito, S. *et al.* Single-nucleus chromatin accessibility and transcriptomic characterization of Alzheimer's disease. *Nature genetics* **53**, 1143–1155 (2021).
54. Sadick, J. S. *et al.* Astrocytes and oligodendrocytes undergo subtype-specific transcriptional changes in Alzheimer's disease. *Neuron* **110**, 1788–1805 (2022).
55. Smith, A. M. *et al.* Diverse human astrocyte and microglial transcriptional responses to Alzheimer's pathology. *Acta Neuropathologica* **143**, 75–91 (2022).

56. Wang, Q. *et al.* Single-cell transcriptomic atlas of the human substantia nigra in Parkinson's disease. *Biorxiv* 2022–03 (2022).
57. Yang, A. C. *et al.* A human brain vascular atlas reveals diverse mediators of Alzheimer's risk. *Nature* **603**, 885–892 (2022).
58. Zhang, L. *et al.* Single-cell transcriptomic atlas of Alzheimer's disease middle temporal gyrus reveals region, cell type and sex specificity of gene expression with novel genetic risk for MERTK in female. *medRxiv* 2023–02 (2023).
59. Zhu, B. *et al.* Single-cell transcriptomic and proteomic analysis of Parkinson's disease brains. *Science Translational Medicine* **16**, eabo1997 (2024).
60. Wolf, F. A., Angerer, P. & Theis, F. J. SCANPY: Large-scale single-cell gene expression data analysis. *Genome biology* **19**, 1–5 (2018).
61. Lause, J., Berens, P. & Kobak, D. Analytic Pearson residuals for normalization of single-cell RNA-seq UMI data. *Genome biology* **22**, 1–20 (2021).
62. Korsunsky, I. *et al.* Fast, sensitive and accurate integration of single-cell data with Harmony. *Nature methods* **16**, 1289–1296 (2019).

**Correspondence** Correspondence to Jong Chul Ye or Inkyung Jung.

**Acknowledgements** This research was supported by National Research Foundation of Korea(NRF) (\*\*RS-2023-00262527\*\*).

**Author Contributions** These authors contributed equally: Gyutaek Oh, Baekgyu Choi.  
G.O. developed the code, conducted experiments, analyzed the results, and drafted and revised the manuscript.

B.C. collected the data, analyzed the results, and drafted and revised the manuscript. S.J. analyzed the results and revised the manuscript. J.C.Y. and I.J. supervised the project, guided its conceptualization and discussions, and prepared the manuscript.

**Competing Interests** The Authors declare no competing interests.

## Supplementary Information

Supplementary Table 1: Hyperparameters for pre-training and fine-tuning.

Task	Pre-Training	Fine-Tuning
Learning Rate	$1e^{-3}$	$1e^{-4}$
Training Epoch	2	5, 10
Optimizer	AdamW	
Optimizer Momentum	$\beta_1, \beta_2 = 0.9, 0.95$	
Weight Decay	0.1	
Learning Rate Scheduler	Cosine decay	
Batch Size	16	

**Supplementary Table 2: The distribution of cell types in the datasets (AC: astrocyte, MG: microglia, OL: oligodendrocyte, OPC: oligodendrocyte progenitor cell, EXN: excitatory neuron, INN: inhibitory neuron, EC: endothelial cell, PC: pericyte, DT: doublet, ETC: others).**

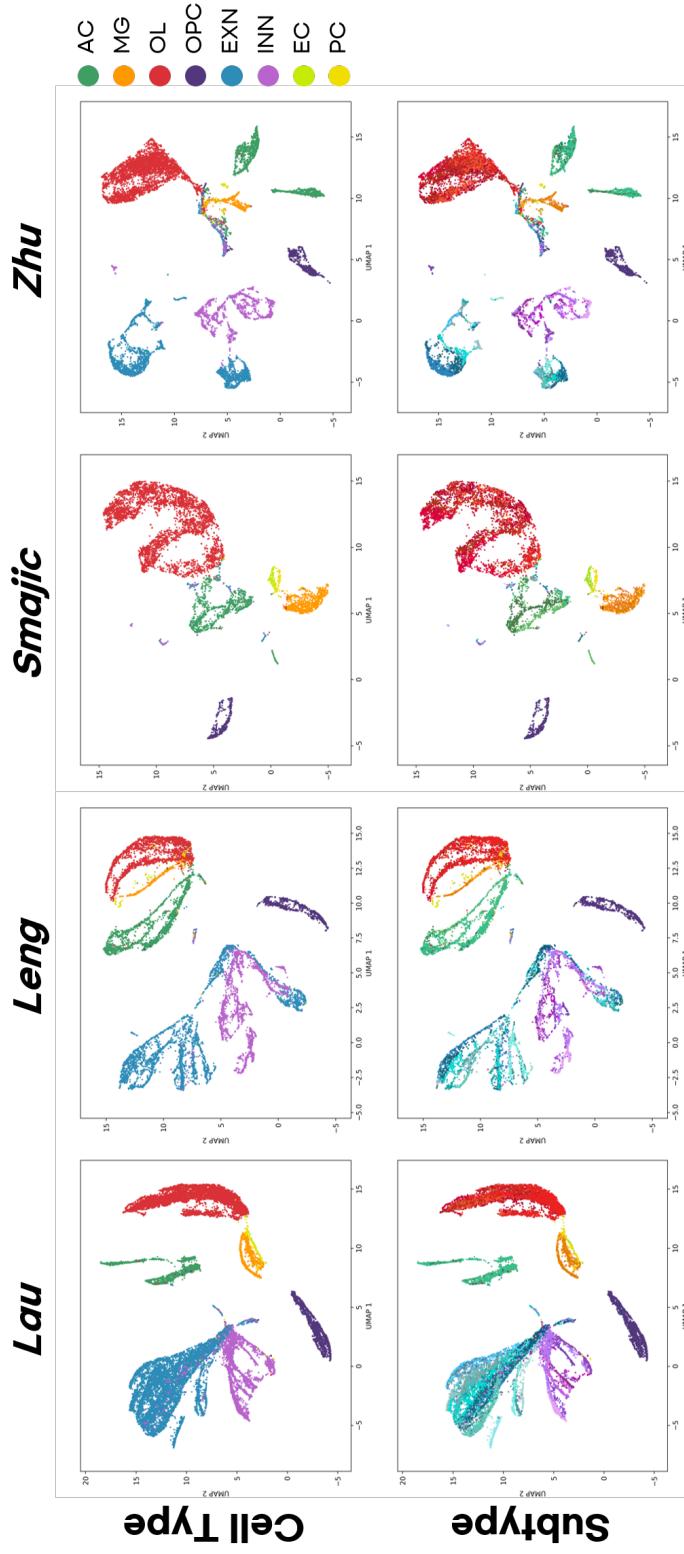
Task	Dataset	AC	MG	OL	OPC	EXN	INN	EC	PC	DT	ETC	Total (Train/Validation/Test)
Pre-Training	<i>Agarwal</i>	415	273	3,933	413	6,557	2,734	14	10	1,703	588	16,640 (16,640/0/0)
	<i>Gerrits</i>	113,506	127,022	35,747	679	4,070	2,854	20,495	7,965	37,107	28,154	377,599 (377,599/0/0)
	<i>Kamath</i>	40,848	34,816	185,451	15,148	28,031	11,570	5,786	3,927	5,603	99,132	430,312 (430,312/0/0)
	<i>Marcos</i>	385	11	169	69	82,452	13,782	25	13	66	4,175	101,147 (101,147/0/0)
	<i>Morabito</i>	4,253	3,600	36,551	2,567	5,775	5,524	126	174	1,179	269	60,018 (60,018/0/0)
	<i>Sadick</i>	52,299	3,955	35,231	3,148	13,163	6,405	512	1,470	4,599	3,869	124,651 (124,651/0/0)
	<i>Smith</i>	55,351	25,766	388	1,230	1,208	2,390	924	451	3,504	2,173	93,385 (93,385/0/0)
	<i>Wang</i>	5,942	7,803	81,378	8,361	5,987	840	4,804	2,137	21,202	6,716	145,170 (145,170/0)
	<i>Yang</i>	22,295	2,190	30,372	2,477	720	1,693	38,293	23,955	38,176	3,043	163,214 (163,214/0/0)
	<i>Zhang</i>	6,039	3,807	22,245	4,837	20,429	10,328	171	185	4,036	723	72,800 (72,800/0/0)
Downstream Tasks	<i>Lau</i>	12,157	4,719	30,571	9,223	50,572	18,932	540	444	12,427	-	139,585 (96,311/12,522/30,752)
	<i>Leng</i>	6,650	2,260	11,904	3,038	19,926	9,656	194	135	3,590	-	57,353 (39,911/5,059/12,383)
	<i>Smajic</i>	5,018	3,717	20,956	2,674	980	705	1,641	673	1,479	-	37,843 (26,929/3,692/7,222)
	<i>Zhu</i>	7,077	4,386	22,773	4,737	19,200	11,763	233	156	3,425	-	73,750 (52,753/6,835/14,162)
<i>Jung</i>	<i>Jung</i>	36,605	16,860	144,237	17,993	121,246	43,161	9,897	6,463	54,164	-	450,626 (323,754/36,095/90,777)

Supplementary Table 3: The distribution of disease in the datasets (CN: cognitively normal, AD: Alzheimer’s disease, PD: Parkinson’s disease, LB: Lewy body dementia, ETC: unknown).

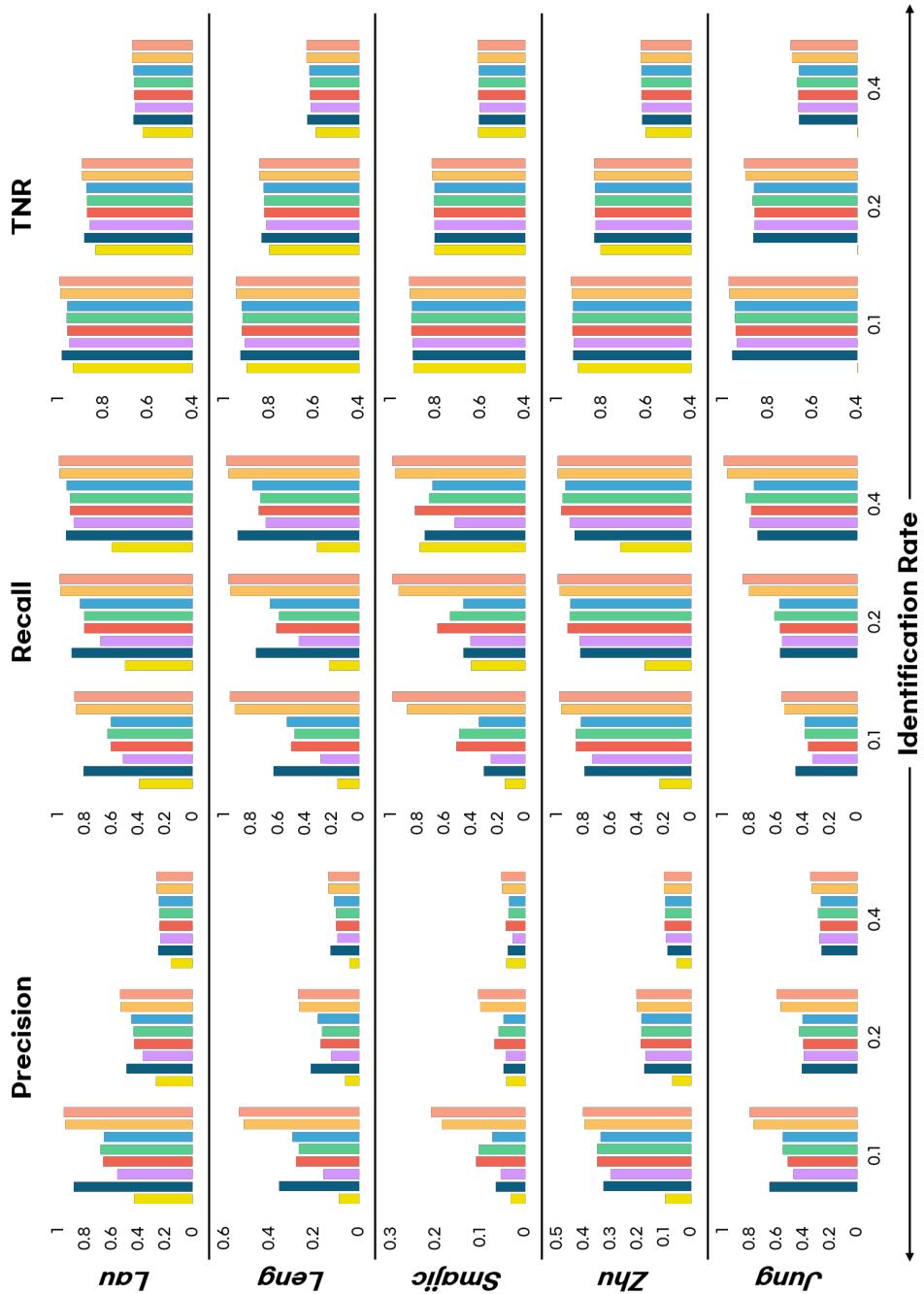
Task	Dataset	CN	AD	PD	LB	ETC	Total (Train/Validation/Test)
Pre-Training	<i>Agarwal</i>	12	0	0	0	0	12 (12/0/0)
	<i>Gerrits</i>	16	20	0	0	0	36 (36/0/0)
	<i>Kamath</i>	44	0	32	18	3	97 (97/0/0)
	<i>Marcos</i>	8	19	0	0	0	27 (27/0/0)
	<i>Morabito</i>	7	11	0	0	0	18 (18/0/0)
	<i>Sadick</i>	10	14	0	0	0	24 (24/0/0)
	<i>Smith</i>	12	12	0	0	0	24 (24/0/0)
	<i>Wang</i>	9	0	19	0	5	33 (33/0)
	<i>Yang</i>	12	13	0	0	0	25 (25/0/0)
	<i>Zhang</i>	6	6	0	0	0	12 (12/0/0)
Downstream Tasks	<i>Lau</i>	9	12	0	0	0	21 (15/2/4)
	<i>Leng</i>	6	14	0	0	0	20 (14/2/4)
	<i>Smajic</i>	6	0	5	0	0	11 (7/2/2)
	<i>Zhu</i>	6	0	6	0	0	12 (8/2/2)
	<i>Jung</i>	9	8	5	3	0	25 (15/4/6)

**Supplementary Table 4:** Average F1 scores for cell type, subtype, and subcluster classification of various methods.

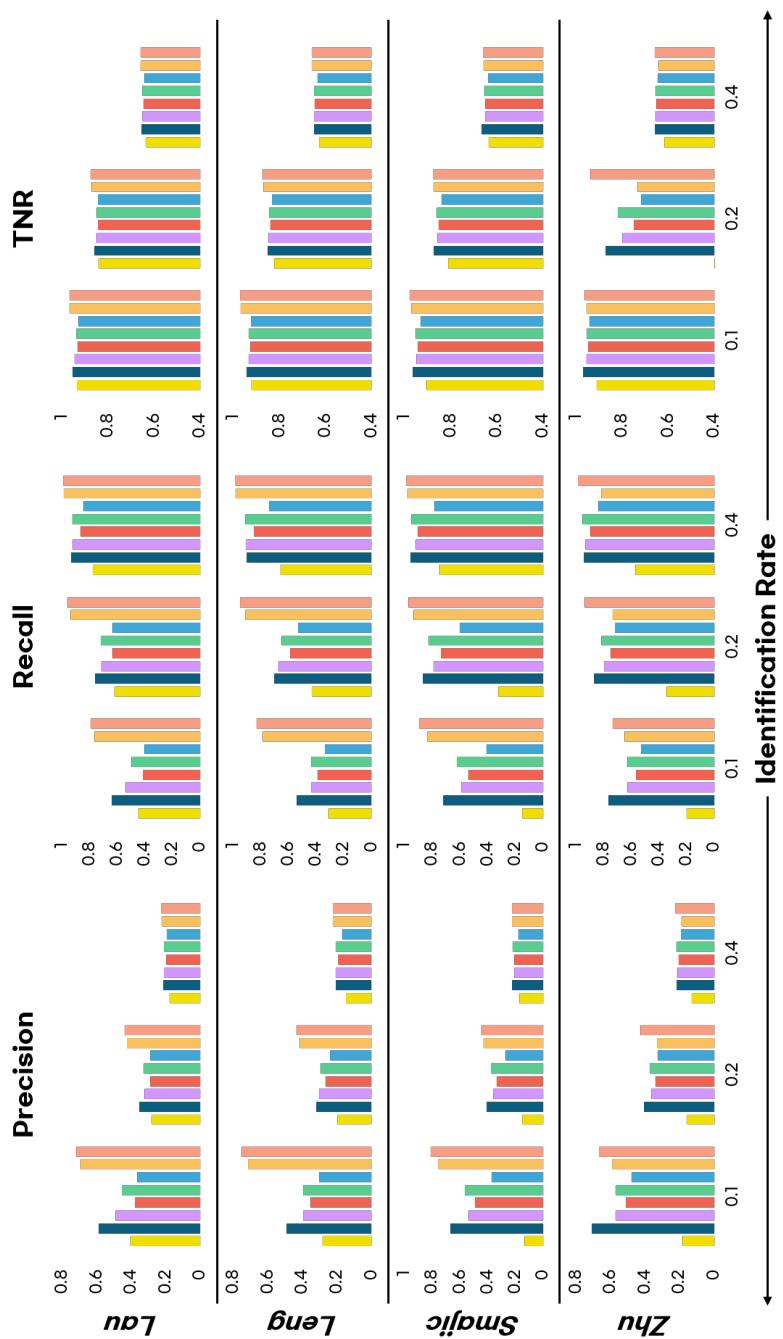
Data	Method	F1 Score (Cell Type)			F1 Score (Subtype)			F1 Score (Subcluster)		
		Macro	Micro	Weighted	Macro	Micro	Weighted	Macro	Micro	Weighted
<i>Lau</i>	Seurat	0.9938	0.9953	0.9953	0.4162	0.6305	0.6219	-	-	-
	SciBet	0.9943	0.9968	0.9968	0.6557	0.7437	0.7453	0.5206	0.6486	0.6497
	scBERT	0.9939	0.9978	0.9978	0.6457	0.7895	0.7838	0.5034	0.6975	0.6907
	scHyena	<u>0.9960</u>	<b>0.9982</b>	<b>0.9982</b>	0.5352	0.7800	0.7696	0.3983	0.6579	0.6430
	Proposed	<b>0.9983</b>	<b>0.9982</b>	<b>0.9982</b>	<b>0.6916</b>	<b>0.8168</b>	<b>0.8137</b>	<b>0.5575</b>	<b>0.7425</b>	<b>0.7402</b>
<i>Leng</i>	Seurat	0.8248	0.9936	0.9926	0.4088	0.5649	0.5548	-	-	-
	SciBet	0.9905	0.9965	0.9965	0.6194	0.7069	0.6969	0.4843	0.5791	0.5780
	scBERT	0.9800	0.9961	0.9960	0.6027	<b>0.7703</b>	0.7588	0.4549	0.6638	0.6456
	scHyena	0.9796	<u>0.9970</u>	<u>0.9970</u>	0.4131	0.6855	0.6615	0.2972	0.5921	0.5590
	Proposed	<b>0.9947</b>	<b>0.9976</b>	<b>0.9976</b>	<b>0.6503</b>	<u>0.7641</u>	<b>0.7595</b>	<b>0.5203</b>	<b>0.6669</b>	<b>0.6610</b>
<i>Smajic</i>	Seurat	0.9426	0.9922	0.9920	0.3016	0.5413	0.5351	-	-	-
	SciBet	0.9481	0.9946	0.9941	<b>0.4738</b>	0.6888	0.6934	0.3557	0.6068	0.6040
	scBERT	0.9532	0.9950	0.9946	0.2920	0.7383	0.7052	0.1652	0.5807	0.5399
	scHyena	<b>0.9843</b>	<b>0.9979</b>	<b>0.9979</b>	0.2953	<u>0.7549</u>	<u>0.7384</u>	0.2145	<u>0.6702</u>	<u>0.6498</u>
	Proposed	<u>0.9742</u>	<u>0.9967</u>	<u>0.9968</u>	<u>0.3807</u>	<b>0.8047</b>	<b>0.7948</b>	<b>0.3739</b>	<b>0.7396</b>	<b>0.7395</b>
<i>Zhu</i>	Seurat	0.9921	0.9962	0.9962	0.5374	0.6289	0.6265	-	-	-
	SciBet	0.9892	0.9943	0.9943	0.6805	0.6160	0.6389	0.4881	0.5153	0.5351
	scBERT	0.9529	0.9938	0.9937	0.5703	<u>0.7406</u>	<u>0.7139</u>	0.3606	0.5754	0.5442
	scHyena	0.9880	<u>0.9968</u>	<u>0.9968</u>	0.5183	0.7280	0.7103	0.3439	<u>0.6269</u>	<u>0.6016</u>
	Proposed	<b>0.9940</b>	<b>0.9974</b>	<b>0.9974</b>	<b>0.6889</b>	<b>0.7770</b>	<b>0.7775</b>	<b>0.5667</b>	<b>0.7025</b>	<b>0.7008</b>
<i>Jung</i>	Seurat	0.9245	0.9209	0.9227	0.3002	0.4478	0.4646	-	-	-
	SciBet	0.9455	0.9452	0.9467	0.6466	0.5948	0.5968	0.4898	0.4805	0.4896
	scBERT	0.9747	<b>0.9791</b>	<b>0.9791</b>	<u>0.6900</u>	<u>0.7028</u>	<u>0.7038</u>	<u>0.5205</u>	<u>0.6161</u>	<u>0.6069</u>
	scHyena	0.9751	<u>0.9790</u>	<u>0.9790</u>	0.6142	0.6840	0.6752	0.4765	0.6111	0.5938
	Proposed	<b>0.9772</b>	0.9766	0.9768	<b>0.7135</b>	<b>0.7168</b>	<b>0.7194</b>	<b>0.5911</b>	<b>0.6587</b>	<b>0.6578</b>



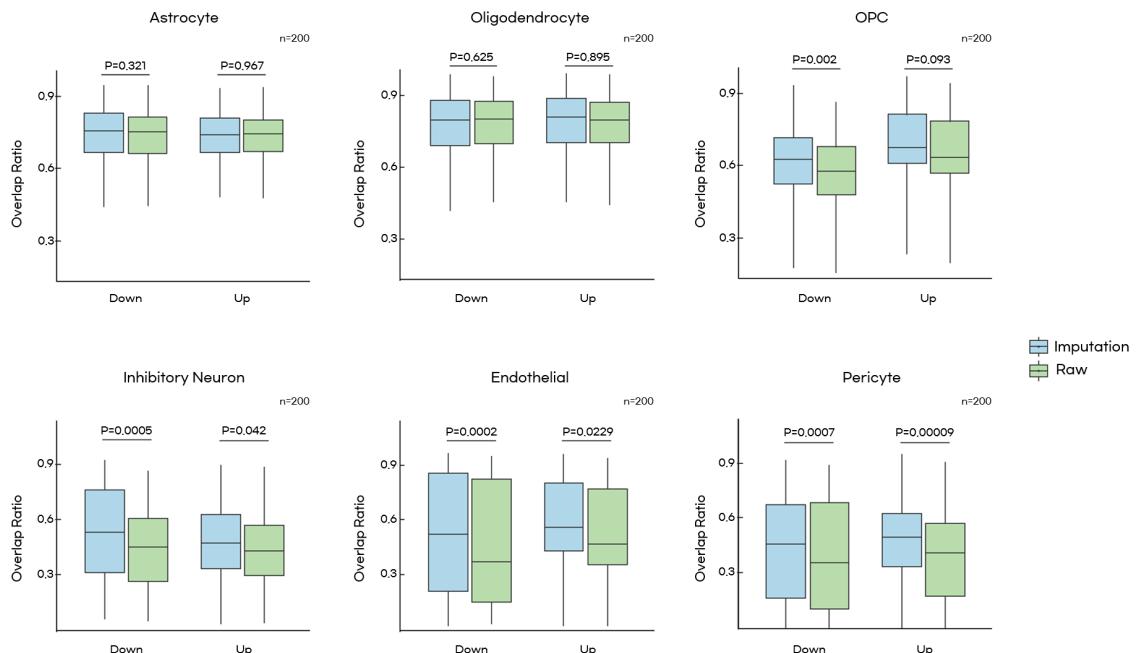
**Supplementary Fig. 1:** UMAP visualization of cell embeddings from the pre-trained scMamba model. Each UMAP is colored based on 8 major cell types or 72 subtypes. (AC: astrocyte, MG: microglia, OL: oligodendrocyte, OPC: oligodendrocyte progenitor cell, EXN: excitatory neuron, INN: inhibitory neuron, EC: endothelial cell, PC: pericyte).



Supplementary Fig. 2: Precision, recall, and TNR for in vivo doublet detection by each method across datasets. The x-axis of the histograms represents the identification rate.



Supplementary Fig. 3: Precision, recall, and TNR for simulated doublet detection by each method across datasets. The x-axis of the histograms represents the identification rate.



**Supplementary Fig. 4:** Boxplots showing for the fraction of DEG overlap for six cell types after selecting half of samples within the same study. The number of cells for each cell type is shown together: Astrocytes = 5,018, Oligodendrocytes = 20,956, OPCs = 2,674, Inhibitory neurons = 705, Endothelial cells = 1,641, and Pericytes = 673. P-values were calculated using paired t-test.